Efectos de los Patrones de Aprendizaje en Línea sobre el Rendimiento Académico desde una Perspectiva de la Minería de Datos

Universidad de Oviedo

Moisés Riestra González

Directores

Dra. María del Puerto Paule Ruíz y Dr. Francisco Ortín Soler

Departamento de Informática

Universidad de Oviedo

Una investigación realizada para el programa de doctorado

Informática

Oviedo, España

Mayo 2022

Universidad de Oviedo

RESUMEN DEL CONTENIDO DE TESIS DOCTORAL

1.- Título de la Tesis	
Español/Otro Idioma: **Efectos de los patrones de aprendizaje en línea sobre el rendimiento académico desde una perspectiva de la minería de datos**	Inglés: **Effects of online behaviors on academic performance from a data mining perspective**

2.- Autor	
Nombre: **Moisés Riestra González**	DNI/Pasaporte/NIE: [redacted]
Programa de Doctorado: **Ingeniería Informática**	
Órgano responsable: **Centro Internacional de Postgrado**	

RESUMEN (en español)

La predicción temprana del rendimiento de los estudiantes es un recurso valioso para mejorar su aprendizaje. Ser capaces de detectar alumnos en riesgo o excelentes en las etapas iniciales del curso permitiría disponer de más tiempo para tomar acciones orientadas a mejorar o incentivar su rendimiento respectivamente. Idealmente, la predicción a realizar no debería ser específica de una asignatura, metodología, área de conocimiento, duración o sistema de evaluación, pudiendo así ser aplicada a un número variado de asignaturas.

Existen estudios orientados a predecir el rendimiento de los estudiantes mediante el análisis de archivos de registro de las interacciones que realizan los alumnos dentro de las plataformas de aprendizaje (Learning Management System, LMS). La mayoría de los trabajos crean modelos predictivos a partir de los registros de las interacciones de los alumnos al finalizar el curso académico. Sin embargo, estos modelos no son útiles para la predicción temprana, porque la información utilizada para entrenar los modelos no es la misma que la empleada en fase de inferencia.

Algunos estudios sí abordan la predicción mediante el entrenamiento con datos tomados únicamente de etapas iniciales de las asignaturas. No obstante, la principal carencia de estos trabajos es que únicamente se centran en un tipo particular de asignatura. Esto hace que los modelos creados no sean reutilizables en otras asignaturas, además, en muchos casos poseen un rendimiento inferior.

En esta tesis doctoral utilizamos algoritmos de aprendizaje automático para crear modelos de predicción temprana del rendimiento de los alumnos, utilizando la información de registro de los LMSs de todas las asignaturas de un año académico en la Universidad de Oviedo. Estos modelos únicamente utilizan acciones realizadas por los estudiantes hasta el momento en el que se realiza el análisis, empleando así la misma información tanto en la fase de entrenamiento como de inferencia.

Creamos modelos con la información generada previamente a los instantes 10%, 25%, 33% y 50% de la duración total de la asignatura. El objetivo no es predecir la calificación numérica de los estudiantes, sino detectar alumnos en riesgo, suspensos y excelentes en las primeras etapas de la asignatura. El rendimiento de los modelos aumenta a medida que aumenta el momento de la predicción, desde el 80,1% cuando solo ha transcurrido un 10% de la asignatura hasta un 90% cuando se ha impartido la mitad de ella.

Adicionalmente, realizamos un análisis de los modelos predictivos para saber qué variables son las más influyentes en la predicción del rendimiento. Dicho análisis se realiza sobre los modelos basados en árboles de decisión, al facilitar estos la interpretabilidad y explicabilidad, además de ofrecer un elevado rendimiento. Este análisis permite comprobar cómo distintas variables influyen en el rendimiento conforme el instante de predicción de los modelos avanza.

Además de la utilización de algoritmos de aprendizaje automático supervisado, también se han utilizado algoritmos no supervisados de agrupamiento para la obtención de patrones de comportamiento con respecto a la interacción de los alumnos con el LMS. Con este análisis hemos identificado seis patrones de comportamiento que se repiten a lo largo de los distintos instantes de análisis. Adicionalmente, se ha comprobado que cuatro de los grupos obtenidos

tienen una correlación alta con el rendimiento de los alumnos en la plataforma de aprendizaje.
Otro trabajo de investigación realizado se basa en el análisis exhaustivo de las acciones de los alumnos para una asignatura en particular de la que poseíamos información adicional de evaluación del alumnado. Para ello se utilizaron reglas de asociación para ver qué variables estaban asociadas al rendimiento de los alumnos. La principal conclusión obtenida del trabajo es que la procrastinación del estudiantado está relacionada con un menor rendimiento.

RESUMEN (en inglés)

Early student performance prediction is a valuable resource for improving student learning. Being able to detect students at risk in the initial stages of the course would allow teachers and students more time to take actions aimed at improving their performance. Also, the early detection of excellent students could be used to propose additional personalized activities so we can increase their motivation. The prediction should not be specific to methodology, area of knowledge, duration, or evaluation system of the course, so the same approach can be applied to a varied number of courses.
There are previous studies focused on predicting student performance by analyzing log files which contain all the actions performed by students within learning platforms (Learning Management System, LMS). Most of the studies create predictive models from the records of student interactions at the end of the academic year. However, these models are not useful for early prediction, because the information used to train the models is not the same as that used in the inference phase.
A few previous studies carried out the prediction through training with data taken only from the initial stages of the courses. However, the main shortcoming of these works is that they only focus on a particular type of course. This means that the created models are valid for the same courses for which they were trained, but do not have the desired performance for another types.
In this thesis we use machine learning algorithms to create course-agnostic models for early prediction of student performance, using information for all the course of an academic year at the University of Oviedo. For that we only use the log information of learning platforms. These models only use actions performed by the students up to the moment in which the analysis is carried out, thus using the same information both in the training phase and in the inference phase.
We create models with the information generated when the duration of the course has been 10 %, 25 %, 33% and 50% of the total duration of them. The objective is not to predict the exact numerical grade of the students, but to detect at-risk, failing, and excellent students in the early stages of the course when it is still possible to modify the learning process. The performance of the models increases as the prediction moment increases. The accuracy increases from 80.1% when only 10% of the course has passed to 90% when the prediction moment is the half of the course.
Furthermore, we perform an analysis of the prediction models to find out which variables are the most influential in predicting student performance. White box analysis is carried out on models obtained with the algorithm decision trees. The results of this algorithm facilitate the explainability and interpretability of the created models. This analysis allows us to understand how different variables influence in the performance of the students as the prediction moment of the models progresses.
In addition to training models based on supervised machine learning algorithms, unsupervised clustering algorithms have also been used to obtain behavioral patterns regarding the interaction of students with the LMS. With this analysis we have identified six behavior patterns. The same six groups appear in the four different moments of analysis. It also has been verified that four of the groups obtained have a high correlation with the performance of the students in the learning platform.
Another research work carried out as part of this investigation is based on the exhaustive analysis of the actions of the students for a particular course for which we had additional information on the evaluation of the students. To do this, association rules were used to see which variables were associated with student performance. The main conclusion obtained from the work is that student procrastination is related to an evaluation below the average.

Universidad de Oviedo

SR. PRESIDENTE DE LA COMISIÓN ACADÉMICA DEL PROGRAMA DE DOCTORADO EN __INFORMÁTICA_____

Resumen

La predicción temprana del rendimiento de los estudiantes es un recurso valioso para mejorar su aprendizaje. Ser capaces de detectar alumnos en riesgo en las etapas iniciales del curso permitiría disponer de más tiempo para tomar acciones orientadas a mejorar su rendimiento. Asimismo, la detección temprana de estudiantes excelentes podría utilizarse para proponerles actividades adicionales personalizadas y aumentar así su motivación. Idealmente, la predicción a realizar no debería ser específica de una asignatura, metodología, área de conocimiento, duración o sistema de evaluación, pudiendo así ser aplicada a un número variado de asignaturas.

Existen estudios orientados a predecir el rendimiento de los estudiantes mediante el análisis de archivos de registro (*log*) de las acciones e interacciones que realizan los alumnos dentro de las plataformas de aprendizaje (*Learning Management System*, LMS). La mayoría de los trabajos crean modelos predictivos a partir de los registros de las interacciones de los alumnos al finalizar el curso académico. Sin embargo, estos modelos no son útiles para la predicción temprana, porque la información utilizada (archivos de registro o *log*) para entrenar los modelos no es la misma que la empleada en fase de inferencia, donde en función del instante de predicción se dispone de pocos datos para predecir el rendimiento del alumnado.

Algunos pocos estudios sí abordan la predicción mediante el entrenamiento con datos tomados únicamente de etapas iniciales de las asignaturas. No obstante, la principal carencia de estos trabajos es que únicamente se centran en un tipo particular de asignatura. Esto hace que los modelos creados sean válidos para las asignaturas para los que fueron entrenados, pero no posean el rendimiento deseado para otro tipo de asignaturas.

En esta tesis doctoral utilizamos algoritmos de aprendizaje automático (*machine learning*, ML) para crear modelos de predicción temprana del rendimiento de los alumnos, con tan solo utilizar la información de registro (ficheros *log*) de plataformas de aprendizaje. Estos modelos únicamente utilizan acciones realizadas por los estudiantes hasta el momento en el que se realiza el análisis, empleando así la misma información tanto en la fase de entrenamiento como en la de inferencia. Además, los modelos creados son agnósticos respecto a las asignaturas, ya que el conjunto de datos (*dataset*) utilizado para crear y validar los modelos contiene información de todas las asignaturas de un año académico en la Universidad de Oviedo.

Creamos modelos con la información generada cuando la duración del curso ha sido del 10 %, 25 %, 33 % y 50 % de la duración total de la asignatura. El objetivo no es predecir la calificación numérica exacta de los estudiantes (regresión), sino detectar alumnos en riesgo, suspensos y excelentes (clasificación) en las primeras etapas de la asignatura, cuando aún es posible modificar el proceso de aprendizaje. Para cada una de las combinaciones de tiempo y grupos de estudiantes descritos, se generan modelos de clasificación utilizando diferentes algoritmos de aprendizaje automático. El rendimiento de los modelos aumenta a medida que aumenta el momento de la predicción. Tales rendimientos van desde el 80,1 % cuando solo ha transcurrido un 10 % de la asignatura hasta un 90 % cuando se ha impartido la mitad de ella.

Adicionalmente, realizamos un análisis de los modelos de predicción para saber qué variables son las más influyentes a la hora de predecir el rendimiento de los alumnos. Dicho análisis de 'caja blanca' se realiza sobre los modelos basados en árboles de decisión, al facilitar la interpretabilidad y explicabilidad de los modelos creados, además de ofrecer un elevado rendimiento. Este análisis permite comprobar cómo distintas variables influyen en el rendimiento conforme el instante de predicción de los modelos avanza.

Además de la utilización de algoritmos de aprendizaje automático supervisado, también se han utilizado algoritmos no supervisados de agrupamiento para la obtención de patrones de comportamiento con respecto a la interacción de los alumnos con el LMS. Con este análisis hemos identificado seis patrones de comportamiento que se repiten a lo largo de los distintos instantes de análisis. Adicionalmente, se ha comprobado que cuatro de los grupos obtenidos tienen una correlación alta con el rendimiento de los alumnos en la plataforma de aprendizaje (*e-learning*).

Otro trabajo de investigación realizado como parte de la presente tesis doctoral se basa en el análisis exhaustivo de las acciones de los alumnos para una asignatura en particular de la que poseíamos información adicional de evaluación del alumnado. Para ello se utilizaron reglas de asociación para ver qué variables estaban asociadas al rendimiento de los alumnos. La principal conclusión obtenida del trabajo es que la procrastinación del estudiantado está relacionada con una evaluación por debajo de la media.

Palabras Clave

Sistemas de aprendizaje a distancia, registros de interacción, predicción del rendimiento del estudiantado, detección temprana, aprendizaje automático, patrones de comportamiento, procrastinación.

Abstract

Early student performance prediction is a valuable resource for improving student learning. Being able to detect students at risk in the initial stages of the course would allow teachers and students more time to take actions aimed at improving their performance. Also, the early detection of excellent students could be used to propose additional personalized activities so we can increase their motivation. The prediction should not be specific to methodology, area of knowledge, duration, or evaluation system of the course, so the same approach can be applied to a varied number of courses.

There are previous studies focused on predicting student performance by analyzing log files which contain all the actions and interactions performed by students within learning platforms (Learning Management System, LMS). Most of the studies create predictive models from the records of student interactions at the end of the academic year. However, these models are not useful for early prediction, because the information used to train the models is not the same as that used in the inference phase. Depending on the prediction instant, the available data to predict student performance could only contains a few actions within the system.

A few previous studies carried out the prediction through training with data taken only from the initial stages of the courses. However, the main shortcoming of these works is that they only focus on a particular type of course. This means that the created models are valid for the same courses for which they were trained, but do not have the desired performance for another types.

In this thesis we use machine learning algorithms (ML) to create models for early prediction of student performance. For that we only use the log information of learning platforms. These models only use actions performed by the students up to the moment in which the analysis is carried out, thus using the same information both in the training phase and in the inference phase. In addition, the models created are course-agnostic, since the dataset used to train and validate the models contains information on all the courses of an academic year at the University of Oviedo.

We create models with the information generated when the duration of the course has been 10 %, 25 %, 33 % and 50 % of the total duration of them. The objective is not to predict the exact numerical grade of the students (regression), but to detect at-risk, failing, and excellent

students (classification) in the early stages of the course when it is still possible to modify the learning process. For each of the described combinations of time and student groups, classification models are generated using different machine learning algorithms. The performance of the models increases as the prediction moment increases. The accuracy increases from 80.1 % when only 10 % of the course has passed to 90 % when the prediction moment is the half of the course.

Furthermore, we perform an analysis of the prediction models to find out which variables are the most influential in predicting student performance (explainability). 'white box' analysis is carried out on models obtained with the algorithm decision trees. The results of this algorithm facilitate the explainability and interpretability of the created models. This analysis allows us to understand how different variables influence in the performance of the students as the prediction moment of the models progresses.

In addition to training models based on supervised machine learning algorithms, unsupervised clustering algorithms have also been used to obtain behavioral patterns regarding the interaction of students with the LMS. With this analysis we have identified six behavior patterns. The same six groups appear in the four different moments of analysis. It also has been verified that four of the groups obtained have a high correlation with the performance of the students in the learning platform (e-learning).

Another research work carried out as part of this investigation is based on the exhaustive analysis of the actions of the students for a particular course for which we had additional information on the evaluation of the students. To do this, association rules were used to see which variables were associated with student performance. The main conclusion obtained from the work is that student procrastination is related to an evaluation below the average.

Keywords

Online Learning Management Systems, interaction logs, student performance prediction, early detection, machine learning, behavior patterns, procrastination.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Ciencia, Innovación y Universidades, bajo el Programa Nacional de Investigación, Desarrollo e Innovación (proyecto RTI2018-099235-B-I00), así como por la Universidad de Oviedo (proyecto GR-2011-0040).

Índice de contenidos

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En este primer capítulo se detalla la motivación para la realización de la investigación, así como las principales contribuciones obtenidas. Adicionalmente, se describe la estructura del presente documento con una breve descripción de cada capítulo.

1.1. Motivación

El incremento del uso de las Tecnologías de la Comunicación y la Información (TIC) ha permitido la aparición de nuevos métodos de enseñanza y aprendizaje donde la presencialidad no es un factor obligatorio. Las instituciones de educación superior han adoptado el uso de las TIC para dar soporte tanto a la enseñanza como al aprendizaje. Uno de los ejemplos más claros es el uso de plataformas de aprendizaje a distancia o *Learning Management Systems* (LMSs), cuyo uso dentro de estas instituciones ha sufrido un incremento significativo en los últimos años [1].

Un LMS se define como un sistema informático (*software*) para la entrega, gestión documental, seguimiento, presentación de informes y administración de cursos educativos y otros programas de aprendizaje, formación o desarrollo [2]. Esta tecnología ofrece, mediante el uso de las capacidades actuales de comunicación a través de Internet, una alternativa que permite transformar el aprendizaje presencial tradicional para convertirlo en un aprendizaje mixto o totalmente a distancia [3].

Adicionalmente, los LMSs pueden ser utilizados como un soporte extra al aprendizaje presencial debido a las funcionalidades que ofrecen este tipo de herramientas: gestión de contenidos de los cursos, canales de comunicación entre alumnos y profesores, capacidades para la entrega de tareas y su evaluación, realización de cuestionarios y la evaluación del alumno [4, 5]. Sin embargo, el uso de estos sistemas tiene como principal inconveniente la pérdida de interacción directa entre los alumnos y entre alumnos y profesores [6]. Esta pérdida de comunicación puede implicar que los alumnos no sean conscientes de si su aprendizaje se está desarrollando del modo esperado.

Las plataformas de aprendizaje generan información de registros o trazas (*logs*) que recogen las acciones que los usuarios realizan al interaccionar con el LMS. Estas acciones incluyen la entrada en la herramienta (*login*), el acceso a documentación interna, la realización de tareas o la interacción en foros de discusión, entre otras. Esta información sobre la interacción de los alumnos ha sido utilizada en los últimos años en diferentes estudios para la generación de modelos predictivos que persiguen analizar el rendimiento de los alumnos en las asignaturas [3], detectar la procrastinación de los estudiantes [7] o incluso agrupar a los alumnos en base a sus diferentes patrones de comportamiento [8].

Los modelos obtenidos en los trabajos mencionados detectan patrones de interacción con los LMSs y correlacionan los patrones descubiertos con la evaluación de los estudiantes. Sin embargo, estos estudios están basados en el uso de las acciones obtenidas de única asignatura individual [8, 9] o en un grupo de asignaturas que comparten una misma estructura y/o metodología [10]. Los resultados de estos estudios dejan la puerta abierta a analizar si es posible encontrar esos patrones de comportamiento para distintos tipos de asignaturas, utilizando para ello datos masivos de asignaturas con diferentes metodologías de enseñanza, disciplinas, áreas de conocimiento, formato de enseñanza, duración y mecanismos de evaluación [11].

Otra característica importante de los modelos predictivos generados a partir de las acciones de los alumnos es su capacidad para realizar la predicción del rendimiento de los estudiantes de forma temprana. Si tal predicción se realiza en los instantes iniciales de una asignatura, detectando por ejemplo los alumnos en riesgo de no aprobar, entonces los profesores podrían utilizar esta información para determinar formas de apoyar en el aprendizaje cuando todavía hay tiempo suficiente [12]. Del mismo modo, la detección de estudiantes excelentes ofrece una vía para ofrecerles actividades extra como motivación al aprendizaje.

No obstante, la mayoría de los trabajos relacionados generan los modelos utilizando la información de todo el curso, no siendo eficaces para la predicción temprana [13]. La construcción de modelos de predicción temprana conlleva que durante el entrenamiento solo deba utilizarse la información obtenida hasta el instante en el que se realiza la predicción [13]. Así, la inferencia se ajustará a la validación del modelo, permitiendo su uso en escenarios reales cuando la información de todo el curso no esté disponible.

La principal contribución de esta investigación es el uso de información masiva almacenada en los registros de un LMS (15.994 estudiantes y 8,5 millones de registros) de múltiples (699) asignaturas heterogéneas (presenciales, mixtas o totalmente a distancia) para predecir de forma temprana el rendimiento de los alumnos, con tan solo conocer la información de interacción del alumno con el LMS hasta el instante de predicción. Los modelos predictivos creados son independientes de características propias de las asignaturas, tales como la duración, el área de conocimiento, el mecanismo de evaluación o la metodología. Estos modelos facilitan el desarrollo de estrategias de intervención temprana, tanto para alumnos en riesgo como para alumnos excelentes, cuando aún existe tiempo para incrementar su rendimiento.

El mismo conjunto de datos también es utilizado para detectar patrones de interacción con el LMS, independientemente del tipo de asignatura. Estos patrones de interacción son analizados para entender cómo se agrupan los estudiantes en base a, por ejemplo, su actividad con los recursos, la evaluación de cuestionarios y tareas, la procrastinación y la participación en foros. Este agrupamiento puede ser usado para la creación de sistemas de aprendizaje adaptativos [14, 15] o sistemas de tutoría inteligentes que, en función de la pertenencia a los distintos grupos, podrían adecuar los contenidos ofrecidos a los alumnos, especialmente en entornos de aprendizaje a distancia [16].

1.2. Contribuciones

Tras indicar la principal contribución de la presente tesis doctoral, pasamos a detallar de forma más concreta las cinco contribuciones u objetivos derivados de la anterior:

1. Predicción temprana del rendimiento académico de un alumno en las tareas evaluables dentro de una plataforma de aprendizaje a distancia. El objetivo no es la predicción de la nota final de una asignatura, sino la evaluación relacionada con las distintas tareas evaluables dentro del LMS para un conjunto de asignaturas de diversas temáticas, metodologías, formatos (online, presencial y mixto) y criterios de evaluación. Dicha predicción será obtenida utilizando únicamente las acciones producidas por los alumnos hasta el instante 10 %, 25 %, 33 % y 50 % de la duración de la asignatura. Es importante remarcar que no se realizan predicciones para una asignatura o tipo de asignatura en concreto, sino para un conjunto heterogéneo de ellas. Esto permite que los modelos obtenidos sean aplicables a un gran número y tipo de asignaturas.

2. Explicabilidad de los modelos predictivos. Identificamos las variables independientes más influyentes en los modelos a la hora de predecir el rendimiento de los alumnos para resolver las tareas del LMS. Para ello, se han analizado los resultados de modelos interpretables para cuantificar la importancia de las variables independientes a la hora de predecir el rendimiento de los alumnos en los distintos instantes de tiempo y de la clasificación de la evaluación (calificaciones excelentes, deficientes, aprobados y suspensos).

3. Obtención y análisis del agrupamiento de los alumnos en función de los patrones de utilización de las plataformas de aprendizaje, independientemente de la asignatura. El análisis de los grupos obtenidos indica que existen seis grupos de alumnos diferentes en relación con la utilización del LMS. Estos grupos se repiten en los distintos instantes tempranos en los que se realiza el análisis. Por último, se analizan las variables que definen cada uno de los grupos.

4. Análisis de la correlación entre la pertenencia a un grupo y el rendimiento de los alumnos. Cuatro de los seis grupos encontrados guardan una relación con la evaluación que obtienen los alumnos pertenecientes a dichos grupos. Las correlaciones encontradas son válidas para asignaturas de diferentes

áreas, metodologías, duraciones y formas de aprendizaje.

5. Detección de comportamientos de procrastinación a través de la interacción con un LMS y su asociación con la evaluación real obtenida por los estudiantes, para una asignatura dada. La utilización de reglas de asociación ha permitido descubrir que valores altos de procrastinación (tiempo transcurrido hasta realizar una acción) están asociados al rendimiento bajo del alumnado. Asimismo, un menor tiempo transcurrido se ve asociado a mejores rendimientos. Este estudio, a diferencia del realizado para las cuatro contribuciones anteriores, se ha realizado para una asignatura concreta con una metodología de aprendizaje conocida, definiendo como rendimiento del alumnado las calificaciones finales obtenidas en dicha asignatura.

1.3. Estructura del documento

La investigación presentada en este documento consta de varios capítulos en los que se describe todo el proceso seguido, desde la obtención de los datos hasta la validación de los resultados. A continuación, se describe un pequeño resumen de cada uno de los capítulos en los que se divide el documento.

Este presente Capítulo 1 de introducción describe la motivación del trabajo realizado, así como las principales contribuciones que se han obtenido tras el proceso de investigación llevado a cabo. Asimismo, proporciona la estructura general del documento.

El Capítulo 2 describe los trabajos previos relacionados con la investigación realizada. Éstos están relacionados con la utilización de las acciones de los estudiantes dentro de plataformas de *e-learning* tanto para realizar una predicción del rendimiento de los alumnos como para detectar grupos de alumnos con patrones de comportamiento similares.

El Capítulo 3 versa sobre el preprocesamiento de los datos. En él se describen los datos en crudo almacenados en los registros de la plataforma de aprendizaje y los pasos dados hasta la construcción de un conjunto de datos para el entrenamiento y validación de modelos supervisados y no supervisados.

La metodología utilizada para la construcción de los modelos predictivos mediante aprendizaje automático supervisado está descrita en el Capítulo 4. En dicho capítulo también se incluyen los resultados y las conclusiones obtenidas de los modelos generados, así como la explicación de las variables que son más relevantes a la hora de predecir el rendimiento de los alumnos.

El Capítulo 5 muestra, de una forma similar al anterior, la metodología para la generación de modelos de agrupamiento de los alumnos en relación con su interacción con el LMS, así como los resultados y las conclusiones extraídas de los mismos. También incluye un análisis de correlación entre la pertenencia cada grupo obtenido y la evaluación en las tareas realizadas en la plataforma de *e-learning*.

El Capítulo 6 describe el trabajo que relaciona la procrastinación en los LMSs

y el rendimiento final de los estudiantes, para una asignatura dada. Se incluye la metodología para la obtención de esta relación, así como los resultados y conclusiones obtenidas.

Las principales conclusiones de esta tesis doctoral se presentan en el Capítulo 7. También se incluyen líneas de trabajo futuro en el Capítulo 8 que podrían tomar como base esta investigación, abriendo nuevas líneas derivadas y buscando nuevas aproximaciones de resolución del problema resuelto.

Este documento también proporciona una serie de anexos. En ellos se incluye información que proporciona soporte al resto del documento, pero que por su naturaleza de alto detalle dificultaría la lectura fluida del documento.

El Anexo A y el Anexo B incluyen los listados de las variables generadas para los modelos de aprendizaje automático clasificadas en función de su naturaleza, variables de acciones y de evaluación respectivamente; el Anexo C contiene los resultados de la búsqueda de los mejores hiperparámetros en los modelos predictivos; por otra parte, el Anexo D contiene la agrupación de las variables durante la reducción de dimensionalidad para los modelos de agrupamiento; las reglas de asociación obtenidas con los distintos algoritmos en uno de los estudios se muestran en el Anexo E; por último, las publicaciones relacionadas con esta investigación se incluyen en el Anexo F.

Capítulo 2

Trabajo relacionado

El trabajo relacionado con esta tesis doctoral se ha clasificado en dos grupos. Por un lado, se analizan los trabajos de investigación que utilizan aprendizaje automático (*machine learning*) supervisado para predecir el rendimiento académico de los alumnos, utilizando la información de interacción con la plataforma de aprendizaje. El segundo grupo de trabajos se centra en el uso de aprendizaje automático no supervisado para el descubrimiento de patrones de interacción con el LMS durante el proceso de aprendizaje, obteniendo información de los grupos (*clusters*) con comportamientos similares.

Ambos tipos de trabajos están basados en el análisis de los registros de las acciones que realizan los estudiantes con las plataformas de aprendizaje. Esta información almacenada cuenta con un alto nivel de granularidad, describiendo las acciones con un alto grado de detalle [17]. El conjunto de datos no es únicamente un conteo de páginas o recursos vistos, sino que almacena otra información como, por ejemplo, los tiempos empleados para realizar ciertas tareas y las evaluaciones de las tareas realizadas. En líneas generales estas acciones están almacenadas en algún formato tabular dentro de la plataforma de aprendizaje.

2.1. Predicción del rendimiento de los alumnos

Como se ha descrito, el primer grupo de trabajos relacionados está centrado en la creación de modelos predictivos para predecir el rendimiento de los alumnos. En el análisis de este tipo de trabajos, consideramos dos variables fundamentales que hacen nuestro trabajo sea novedoso: el número de asignaturas utilizadas y el instante de predicción (cuánta información de los registros es utilizada para la construcción de los modelos).

2.1.1. Predicción del rendimiento tras finalizar la asignatura

Las investigaciones descritas en esta sección utilizan como conjunto de entrada el conjunto de acciones realizadas por los alumnos en la duración total de las asignaturas. Los modelos que siguen este enfoque permiten la predicción del

rendimiento de forma precisa una vez la asignatura ha concluido, poseyendo poco rendimiento cuando, en fase de inferencia, se usa únicamente la información de registro en instantes tempranos del desarrollo del curso [13].

Predicción del rendimiento tras finalizar una única asignatura

Los estudios descritos en esta sección utilizan datos de una única asignatura en un curso académico o de una única asignatura durante varios años consecutivos, donde no se ha modificado ni la estructura ni los contenidos.

Macfadyen y Dawson analizaron cuán útil es el uso de la información que proporcionan las plataformas LMS para predecir el rendimiento académico de los alumnos [12]. Su investigación estaba basada en las acciones realizadas dentro de una asignatura del grado de Biología, impartida de forma totalmente virtual por la Universidad British Columbia durante el año 2008. Los datos extraídos de la plataforma de *e-learning* incluían frecuencias de acceso a los recursos almacenados en la misma, así como a foros, tareas online y sistemas de gestión. Adicionalmente utilizaron datos del tiempo empleado por los alumnos en ciertos componentes de la plataforma, como las tareas, o el tiempo total de uso de la plataforma. Estas variables fueron utilizadas como entrada a dos modelos predictivos que permitían la predicción de la evaluación de los alumnos y, de esta forma, identificar aquellos que estuviesen en riesgo de no superar la asignatura. Los algoritmos seleccionados en esta investigación fueron una regresión lineal múltiple y una regresión logística binaria. Los autores obtuvieron más del 30 % de explicabilidad de las evaluaciones de los alumnos con el primer modelo, mientras que con el segundo se consiguieron identificar correctamente el 70,3 % de los alumnos en riesgo.

En otro estudio, Ljubobratovic *et al.* mostraron cómo diferentes modelos basados en el algoritmo de clasificación *random forest* fueron capaces de predecir el éxito de los alumnos en una asignatura [18]. El conjunto de variables estaba formado por métricas obtenidas de los cuestionarios, tareas, visualización de vídeos y recursos proporcionados dentro de la herramienta, cuya información está almacenada en los registros de Moodle. En su investigación, utilizaron 408 registros obtenidos de 5 años académicos distintos de la misma asignatura, Programación II, impartida en la Universidad de Rijeka, manteniéndose invariante tanto la estructura como los contenidos a través de los distintos años académicos. Mediante el entrenamiento de un clasificador binario corroboraron que era posible detectar la no superación de una asignatura con un 96,3 % de precisión, si bien encontraron que los resultados no eran fácilmente explicables debido a la naturaleza del algoritmo utilizado. Dado que el algoritmo seleccionado no ofrece explicabilidad, realizaron un análisis ad hoc para conocer qué tipología de variables tenían mayor relevancia, concluyendo que las relacionadas con la evaluación eran las más influyentes en la predicción.

Predicción del rendimiento tras finalizar varias asignaturas

Las investigaciones citadas en la sección anterior utilizaban únicamente información de una asignatura. Las descritas a continuación utilizan variables de diferentes asignaturas, lo que permite la generación de modelos más generalistas

que puedan ser utilizados en un mayor conjunto de asignaturas. No obstante, estos trabajos no consideran instantes iniciales de predicción, creando los modelos con todos los registros generados a lo largo del curso.

La investigación de Romero *et al.* trataba de predecir las evaluaciones que los alumnos universitarios iban a obtener en el examen final en una asignatura [19]. Para ello utilizaban información obtenida directamente de las propias plataformas de aprendizaje, tales como el número de cuestionarios y tareas, tanto realizadas como aprobadas, y el tiempo empleado en los cuestionarios de siete asignaturas de una ingeniería de la Universidad de Córdoba. Para ello compararon la eficiencia de diferentes técnicas de minería de datos y aprendizaje automático, tales como árboles de decisión, redes neuronales, diferentes algoritmos de inducción de reglas como *Apriori* y otros algoritmos como el *k-nearest neighbors* (KNN) para clasificar a los estudiantes. Los mejores resultados fueron obtenidos con árboles de decisión CART, que proporcionaron un rendimiento del 65 %.

Gavsevic *et al.* examinaron la predicción del rendimiento académico en nueve asignaturas dentro de un programa de aprendizaje mixto, donde convivían las clases presenciales y una plataforma de aprendizaje online [20]. Utilizando información de interacción de los alumnos con la plataforma e información institucional de los alumnos, desarrollaron modelos de regresión logística para predecir el estado del aprendizaje (aprobado o suspenso). Para ello utilizaron dos enfoques de generación de modelos predictivos, midiendo la bondad mediante el área bajo la curva ROC (*Receiver Operating Characteristic*) también conocida como AUC (*Area Under the Curve*). El primer enfoque utilizaba un modelo con características de cada una de las asignaturas (modelo generalista) y el resultado obtenido tenía un AUC aceptable ($0{,}5 \leq \text{AUC} \leq 0{,}7$). El segundo enfoque generaba un modelo para cada una de las asignaturas y el AUC se incrementaba considerablemente para todas las asignaturas ($0{,}8 \leq \text{AUC} \leq 1{,}0$). Estos resultados muestran cómo la creación de modelos predictivos para varias asignaturas entraña una mayor dificultad que generarlos para una única asignatura.

Gerritsen utilizó las acciones de los alumnos asociadas a 17 asignaturas almacenadas en los *logs* de Moodle para predecir cuando un alumno iba a superar o no la asignatura [11]. El foco de la investigación estaba situado en la identificación de alumnos en riesgo de no superar la asignatura y que, por tanto, requerían asistencia por parte de los profesores. Para ello seleccionó diferentes algoritmos de clasificación binaria para comprobar cuál de ellos ofrecía una mejor tasa de acierto. El mejor modelo resultó ser una red neuronal con 3 capas ocultas y 16 neuronas en cada capa. La precisión en la clasificación utilizando este modelo alcanzaba un 66,1 %.

López-Zambrano, Lara y Romero estudiaron la portabilidad de modelos de predicción del rendimiento académico de los alumnos entre diferentes asignaturas de un mismo grado y con un mismo uso de la plataforma de aprendizaje [21]. Para ello, utilizaron las acciones registradas en los registros de Moodle de un conjunto de 24 asignaturas, entrenado árboles de decisión J48 para cada una de las asignaturas, de forma individual. La metodología seguida para la comprobación de la portabilidad, además de la generación de los modelos, incluía un agrupamiento

de las asignaturas por temática y por uso de la plataforma de *e-learning*. Para la evaluación de la portabilidad de los modelos utilizaron la métrica de bondad AUC. Cada modelo fue evaluado con asignaturas dentro de su mismo grupo de temática o uso de la plataforma, con el objetivo de medir la mejora o empeoramiento del AUC. Los resultados obtenidos indicaban que portar modelos dentro de áreas de conocimiento similares ofrecía una pérdida de AUC de entre 0,09 y 0,28; mientras que portar a cursos con igual uso de la plataforma implicaba un incremento de AUC entre 0,22 y 0,25.

2.1.2. Predicción temprana del rendimiento

El siguiente grupo de trabajos relacionados se centran en la detección temprana del rendimiento del alumnado y sus posibles aplicaciones dentro del aprendizaje, como la posibilidad de tomar acciones correctivas en instantes iniciales de las asignaturas.

Predicción temprana del rendimiento con una única asignatura

Romero *et al.* utilizaron diferentes técnicas de minería de datos para predecir si un alumno aprueba una asignatura [22]. Para ello emplearon variables cuantitativas y cualitativas sobre la utilización de los foros de discusión por parte de los estudiantes. Dado que parte de la información que consideraban necesaria para el estudio no estaba disponible en los *logs* de Moodle, desarrollaron un *plugin* para capturar esa información adicional. El conjunto de datos contenía información de 114 alumnos de una asignatura del primer curso de Ingeniería Informática. El entrenamiento de modelos se realizó tanto en instantes previos a la finalización de la asignatura como tras su conclusión, utilizando diferentes algoritmos de clasificación como árboles de decisión, *random forest* o *multilayer perceptron* (MLP). De media, los modelos de detección temprana obtenían entre un 70 % y un 80 % de precisión, mientras que esa precisión ascendía a entre el 80 % y el 90 % cuando los datos de la asignatura crecían.

Hu, Lo y Shih analizaron qué precisión podían obtener distintos modelos de aprendizaje automático para predecir el rendimiento de los alumnos en instantes tempranos [13]. Para ello, definieron cuatro tipos de variables en función de las acciones de los estudiantes: (i) acceso a la plataforma, (ii) uso de los materiales de la asignatura, (iii) estado de las tareas, y (iv) participación en los foros de discusión. Los tres instantes definidos para realizar la predicción se correspondían con el final de las semanas 4, 8 y 13 de una asignatura bianual impartida totalmente online, con 330 alumnos matriculados. La generación de los conjuntos de datos para el entrenamiento de modelos en cada instante únicamente tenía en cuenta acciones realizadas previamente al instante. Los resultados del entrenamiento de árboles de decisión CART para la semana 4 proporcionaban un 95 % de precisión, sin mejoras significativas en instantes más avanzados.

La investigación de Marbouti *et al.* se centró en la construcción de tres modelos de regresión logística para la detección temprana de alumnos en riesgo de no superar una asignatura bianual del primer año de Ingeniería [23]. El entrenamiento de estos modelos solo consideraba variables construidas a partir de las acciones

realizadas en las primeras 2, 4 y 9 semanas de la asignatura. Las variables construidas incluían información de asistencia, trabajos realizados, evaluaciones de cuestionarios y evaluaciones de exámenes intermedios. Esta última variable solo se incluía en el conjunto de datos generado tras las 9 primeras semanas. Los resultados de estos modelos permitían identificar alumnos en riesgo con una precisión del 79 % en la semana 2, 90 % en la semana 4 y el 98 % en la semana 9.

En un estudio posterior Marbouti *et al.* utilizaron el mismo enfoque, aplicado esta vez a una asignatura con un sistema de calificación diferente al anglosajón [24]. Esta nueva asignatura utilizaba un sistema de calificación basado en estándares que persiguen medir la calidad de la competencia de los alumnos para conseguir los objetivos definidos en la asignatura [25]. Al igual que en el estudio previo el error cometido en la detección de alumnos en riesgo fue inferior al 10 %.

Costa *et al.* analizaron la eficacia de diferentes algoritmos de aprendizaje automático para la identificación temprana de alumnos en riesgo de no superar la asignatura de Introducción a la Programación en una universidad de Brasil [26]. El estudio estaba divido en dos, debido a que la asignatura tenía dos modalidades: (i) totalmente a distancia con 262 alumnos y 10 semanas de duración, y (ii) presencial con 161 alumnos y 16 semanas de duración. El conjunto de datos para el entrenamiento de los modelos incluía, además de las variables extraídas de las acciones de los estudiantes, variables sociodemográficas como edad y estado civil, entre otras. La métrica seleccionada para medir la bondad de los modelos era el F_1-score, obteniéndose entre el 0,77 y el 0,8 (primera y segunda modalidad, respectivamente) para predecir estudiantes en riesgo en la primera semana de la asignatura, incrementándose esos valores según se incrementaba el momento de predicción.

El *plugin Early Warning System* (EWS) para Moodle tiene como objetivo predecir el rendimiento en la semana 4 del semestre de los alumnos en el primer año de la asignatura *Communication and Information Literacy* de la Universidad de South Pacific [27]. Este *plugin* fue diseñado para capturar información similar a la almacenada en los *logs* de interacciones, como la entrada en la plataforma, completitud de las tareas a realizar y compromiso dentro de la asignatura. Usando como entrada estos datos, Jokhan *et al.* entrenaron un modelo de regresión lineal para predecir la evaluación final numérica de los alumnos, definida como la suma de todas las evaluaciones de la asignatura hasta su finalización en la semana 14. Se obtuvo un R^2 del 0,608, indicando así que el 60,8 % de la variación de la evaluación podía ser explicada con las variables independientes, mediante regresión lineal.

Predicción temprana del rendimiento con múltiples asignaturas

Conijn *et al.* crearon modelos analíticos de predicción de rendimiento académico, usando variables extraídas de las acciones realizadas por 4.989 alumnos en 17 asignaturas presenciales y a distancia [3]. Para ello, utilizaron dos tipologías de modelos: (i) modelos de clasificación binaria (aprobado / suspenso), considerando aprobado si la evaluación era mayor o igual a 5,5; (ii) modelos de regresión para estimar la evaluación del examen final de la asignatura. Vieron cómo la precisión de las predicciones se incrementaba conforme aumentaba el instante de predic-

ción. En la semana 5, la precisión de los modelos de clasificación era del 67 % mientras que el R^2 de los modelos de regresión era del 0,43. Tras realizar una comprobación de las variables más influyentes, se detectó que a partir de esta semana las evaluaciones de tareas intermedias estaban disponibles para realizar la predicción.

La investigación de Olivé *et al.* estaba centrada en predecir si un alumno va a entregar o no las tareas evaluables de la asignatura [28]. Para ello tomaron como entrada las acciones realizadas por 78.722 matriculados en 5.487 asignaturas en los días previos a la fecha de entrega de tareas evaluables. Utilizando el conjunto de datos con las acciones seleccionadas, se entrenaron varias redes neuronales con diferentes arquitecturas y técnicas de selección de las características más influyentes. Los resultados de los distintos modelos ofrecieron precisiones en el rango de 67,46 % a 81,63 % para predecir la entrega de las tareas, con valores F_1 comprendidos entre 71,30 % y 83,09 %.

Tomasevic *et al.* crearon modelos tanto de clasificación como de regresión para predecir la evaluación de los alumnos en un examen final, buscando únicamente comprobar si el alumno iba a aprobar o no [29]. El conjunto de datos utilizado contenía información de dos asignaturas provenientes del *Open University Learning Analytics Dataset* (OULAD). OULAD es un conjunto de datos (*dataset*) que incluye información de las acciones de los alumnos en distintas asignaturas, así como información sociodemográfica asociada a los estudiantes. Los instantes de predicción analizados se correspondían con instantes tras la realización de una tarea intermedia, hasta el instante anterior a la realización del examen final. Los modelos de clasificación incrementaban el valor de la métrica de bondad F_1 desde el 78 % tras la realización de la primera tarea, hasta el 96,6 % justo antes de la realización del examen. El rendimiento de los modelos de regresión mostraba el mismo patrón de mejora, evidenciando que a medida que los modelos disponen de más información eran capaces de realizar mejores predicciones.

2.2. Agrupamiento de los alumnos en base a patrones en las acciones

Los estudios analizados en la sección 2.1 están relacionados con la generación de modelos de aprendizaje automático supervisados para predecir el rendimiento académico de los alumnos. Como hemos comentado, también existen otros estudios que buscan identificar patrones en las acciones de interacción con el LMS para agrupar a los alumnos en base a su comportamiento. Algunos de estos estudios relacionan la pertenencia a un grupo con la evaluación que un alumno obtiene en la asignatura.

Talavera y Gaudioso realizaron varios análisis para identificar los patrones de comportamiento comunes de los estudiantes en función de las acciones que realizan en la plataforma de *e-learning* [30]. El conjunto de variables utilizado para la generación de los grupos incluía: (i) variables generadas a partir de las acciones de una asignatura, almacenadas en los *logs*; (ii) variables sociodemográficas de los alumnos; y (iii) variables relacionadas con los intereses de los alumnos obtenidas

mediante encuestas. Para realizar el agrupamiento se seleccionó el algoritmo *Expectation Maximimization* (EM), obteniendo como resultado seis grupos distintos. Un análisis posterior permitió identificar cierta correlación entre la pertenencia a los grupos y el rendimiento académico.

Hung *et al.* descubrieron patrones de comportamiento en el aprendizaje de asignaturas online los estudiantes de grado en Taiwán, así como cuáles eran los factores de predicción más importantes [31]. Para ello emplearon 17.934 registros correspondientes a las acciones realizadas por 98 estudiantes en la asignatura Aplicaciones de Software Empresarial. Esta asignatura estaba impartida totalmente *online* a través de *Wisdom Mater 2.4*, un LMS ampliamente utilizado en Taiwán. El conjunto de datos de entrada para generar los modelos incluía seis variables: (i) número de accesos a la plataforma; (ii) frecuencia de acceso a los recursos; (iii) número de mensajes escritos en el foro; (iv) número de mensajes leídos en el foro; (v) número de discusiones en las que participa; y (vi) la evaluación del alumno. Utilizando el algoritmo *K-means* se detectaron tres grupos de estudiantes. Dos de los tres grupos agrupaban alumnos con una evaluación por encima de la media y con una interacción con la plataforma alta, mientras que el grupo restante incluía a alumnos por debajo de la media y con poca interacción.

Cobo *et al.* utilizaron *agglomerative hierarchical clustering* para identificar los perfiles de participación de los alumnos en las discusiones dentro de los foros [32]. Las variables utilizadas en este estudio tenían como origen la participación en los foros de discusión durante tres semestres en tres asignaturas diferentes de un Grado en Telecomunicación. A pesar de que las variables independientes eran derivadas de la participación en los foros de discusión, esta participación no era una condición necesaria para la superación de la asignatura. El conjunto de datos contenía información de 672 estudiantes que habían participado en 2.842 discusiones. Como resultado del estudio se identificaron diferentes grupos que se relacionaron con el rendimiento académico: (i) usuarios inactivos (*shirkers*) que no utilizan el foro y están relacionados con un rendimiento muy bajo; (ii) estudiantes lectores (*lurkers*) que leen los foros, pero no postean nueva información, cuyo rendimiento depende de cómo de activos sean, ya que a más actividad mejor rendimiento; y (iii) alumnos que leen y postean (*workers*) que como norma general están relacionados con un buen rendimiento.

La investigación de López, Luna, Romero y Ventura proponía agrupar alumnos en base a su interacción en los foros de discusión de la plataforma de Moodle para predecir su evaluación final [33]. La construcción del conjunto de datos contenía información de la participación en los foros de los alumnos universitarios de primer año. Los grupos obtenidos fueron relacionados con la evaluación que obtenían los alumnos, mostrando cómo, en este escenario, los algoritmos de agrupamiento (*clustering*) conseguían una precisión similar a los modelos supervisados de clasificación.

Cerezo *et al.* generaron grupos a partir de los datos extraídos de los *logs* de Moodle con el objetivo de analizar el proceso de aprendizaje de los estudiantes [8]. Los datos utilizados correspondían a las acciones realizadas por 140 alumnos durante la asignatura Aprendizaje Autónomo de un Grado de Psicología. La asig-

natura estaba dividida en diferentes unidades didácticas, incluyendo cada una de ellas al menos una tarea obligatoria a realizar por los estudiantes. Los algoritmos seleccionados para realizar el agrupamiento fueron *K-means* y EM. Tras realizar el entrenamiento de los modelos no supervisados se obtuvieron cuatro grupos independientes. Posteriormente se realizaron dos estudios ad hoc para analizar tanto las variables más influyentes para el agrupamiento como la relación entre los grupos y la evaluación de los estudiantes. En el primero se detectó que la procrastinación y la socialización dentro de la plataforma eran las variables más influyentes a la hora de agrupar a los alumnos; el segundo estudio identificó cómo tres de los cuatro grupos correlacionaban con el rendimiento académico.

La investigación de Park, Yu y Jo creaba grupos utilizando información de registro LMS no para agrupar alumnos, sino asignaturas [34]. Su '*Latent Class Analysis*' agrupaba asignaturas semipresenciales similares en relación con las acciones que realizan los estudiantes. El estudio utilizó 612 asignaturas de una universidad privada de Corea del Sur. Los resultados obtenidos mostraron la existencia de cuatro grupos de asignaturas diferentes: (i) asignaturas donde no se utilizaba la plataforma (50,5 % del total); (ii) asignaturas con un alto uso de trabajos en grupo y sección de preguntas y respuestas (24,3 %); (iii) asignaturas con un alto uso de lecturas online (18 %); y (iv) asignaturas con un alto uso de recursos y tareas (7,2 %).

Hooshyar, Pedaste y Yang desarrollaron el algoritmo PPP para predecir el rendimiento de los alumnos a función de la procrastinación en la realización de las tareas de Moodle [35]. La asignatura en la que se realizó el estudio constaba de 242 alumnos y pertenecía a la Universidad de Tartu en Estonia. La metodología utilizada constaba de diferentes etapas. Primero, generaban un vector de variables para representar el comportamiento de los alumnos en la entrega de cada una de las tareas. Segundo, obtenían agrupamiento de los alumnos en base a esos vectores, obteniendo tres niveles: procrastinadores, no procrastinadores y potencialmente procrastinadores. Finalmente, se clasificaban los alumnos en base a su evaluación, utilizando diferentes técnicas de clasificación y los resultados del agrupamiento anterior. Se obtuvo 96 % de precisión a la hora de predecir el rendimiento de los alumnos.

Capítulo 3

Preprocesamiento de los datos

La investigación aquí descrita persigue la creación de un conjunto de modelos predictivos y de agrupamiento de los alumnos en función de las acciones que realizan dentro de una plataforma de *e-learning*. Estas acciones están almacenadas en ficheros de registro (*logs*) del LMS en un formato de texto plano y en crudo, no válido para entrenar modelos de forma directa. Debemos, por tanto, procesar esta información para transformar los datos en características o variables independientes (*features*) válidas para obtener el mayor rendimiento predictor posible.

La metodología empleada para la construcción de los conjuntos de datos de los modelos de clasificación y agrupamiento es la misma, utilizando únicamente la información previa al instante de predicción para la generación de las variables necesarias. Es decir, tanto los modelos predictivos como los de agrupamiento se entrenan únicamente con las acciones realizadas por los alumnos previamente a ese instante.

El LMS utilizado en la Universidad de Oviedo es Moodle, uno de los LMSs más utilizados en el aprendizaje semipresencial o totalmente a distancia [36]. Los *logs* de este sistema almacenan la información en crudo de todas las asignaturas impartidas en la Universidad de Oviedo independientemente de la metodología utilizada para su impartición, su duración, la disciplina a la que pertenecen y su formato.

La Figura 3.1 muestra la metodología propuesta para la construcción de las variables que permitan recoger toda la información relevante de las acciones de los estudiantes y que aporte valor para los modelos. El flujo seguido hasta la obtención del conjunto de variables utilizado para entrenar los modelos es el siguiente (todas estas etapas se describen detalladamente en las siguientes secciones):

1. Extracción de información relevante de los *logs*, ya que no todas las acciones son relevantes.
2. Filtrado de aquellas asignaturas donde exista al menos una tarea evaluable.
3. Estimación de la duración de las asignaturas debido a que esta información no está disponible en el sistema.
4. Seleccionar aquellos instantes donde es conveniente realizar la predicción.
5. Generación de las variables independientes únicamente con las acciones has-

ta el instante de predicción seleccionado.
6. Generación de la variable dependiente, usada en los modelos supervisados de predicción del rendimiento académico.

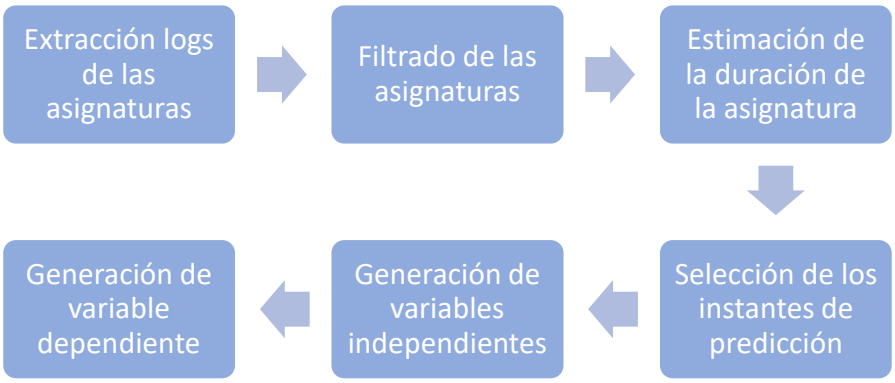

Figura 3.1: Etapas realizadas en el procesamiento de los datos para la generación de conjuntos de datos necesarios para el aprendizaje automático.

El enfoque propuesto en este trabajo consiste en la generación de variables construidas a partir de datos de asignaturas con diferente duración, formato, metodología y pertenecientes a disciplinas distintas. El motivo es conseguir modelos lo suficientemente generalistas para que puedan ser utilizados en cualquier otra asignatura. Tal y como vimos en el Capítulo 2, los modelos construidos a partir de los registros de una única asignatura ofrecen de forma general un rendimiento superior al de los modelos más generalistas. Sin embargo, estos modelos son más difíciles de extrapolar a otras asignaturas donde no se siguen las mismas pautas de comportamiento. Trabajos previos han mostrado una bajada en la precisión de los modelos de entre el 9 % y el 28 % cuando el modelo es usado para la predicción en una asignatura similar, pero para la cual no se había entrenado. Este decrecimiento aumenta a entre el 22 % y el 25 % si se compara con otras asignaturas que tienen un uso similar de la plataforma de *e-learning* [21]. Otro estudio muestra cómo la precisión de los modelos es entre un 60 % y un 80 % superior en modelos realizados con un *dataset* de una única asignatura específica, frente a conjuntos de datos que con información de asignaturas diversas y diferentes, usando la misma información extraída del LMS [20].

3.1. Extracción de *logs* de las asignaturas

Cada uno de los sistemas de *e-learning* disponibles almacenan trazas de las acciones realizadas por los usuarios en sistemas de *logs*. Normalmente los *logs* pueden estar guardados en un conjunto de ficheros o en una o más tablas de una base de datos. En el caso de Moodle, la plataforma utilizada en esta investigación, estas acciones están almacenan dentro de una tabla en la base de datos. Por defecto, las tablas de la base de datos utilizan el prefijo `mdl_` como parte del nombre, si bien esta configuración puede ser modificada en el proceso de instalación. Por ello, los registros de las acciones que realizan los usuarios están almacenados en la tabla `mdl_log`, la cual contiene las siguientes columnas:

- `id`: Campo numérico que se corresponde con un identificador único del registro almacenado. Se corresponde con la clave primaria de la tabla.
- `time`: Campo numérico en formato *UNIX epoch* que indica el momento en el que un usuario ha realizado una acción dentro de la plataforma de aprendizaje. Almacena los segundos transcurridos desde el 1 de enero de 1970 hasta la realización de la acción.
- `userid`: Campo numérico que identifica al usuario que ha realizado la acción en la plataforma.
- `ip`: Campo textual donde se almacena la IP desde la que se realiza la acción en la plataforma. Este campo está anonimizado en el *dataset*. El valor de reemplazo utilizado es 127.0.0.1.
- `course`: Campo numérico que identifica la asignatura en la que un usuario ha realizado la acción.
- `module`: Texto que indica el módulo de la plataforma en el que se realiza la acción. Moodle es una plataforma modular donde cada módulo tiene se propio nombre identificativo. Algunos ejemplos de módulos son *quiz* que representa el módulo de cuestionarios, *assign* para las tareas o *forum* para identificar las acciones realizadas en los foros de discusión.
- `cmid`: Identificador numérico del módulo o recurso sobre el que se realiza la acción.
- `action`: Texto que indica la acción que realiza el usuario. La información almacenada en este campo permite definir que acción se ha realizado dentro de un módulo (campo `module`) en concreto. Las acciones almacenadas son diferentes en función del módulo al que hacen referencia. Por ejemplo, el módulo de cuestionarios dispone acciones de comienzo (*attempt*) y fin de la realización de un cuestionario (*close attempt*), mientras que el módulo de foros de discusión dispone de otras acciones como el acceso al foro (*view forum*) o a un canal de discusión (*view discussion*).
- `URL`: Información textual que muestra el *path* de la URL correspondiente a la acción realizada. Este *path* es relativo a la URL de la plataforma.
- `info`: Información adicional sobre la acción realizada. Los datos almacenados en este campo varían en función de la acción realizada. Por ejemplo, cuando se realiza una acción dentro del módulo de tareas se almacena un texto en HTML que se corresponde con la información mostrada al usuario en la web. En otros casos se almacena un identificador que hace referencia a otras tablas de Moodle en función de la acción realizada. Por último, existen casos como el del módulo de recursos externos (*url*) donde no se almacena información en este campo.
- `stime`: Campo *Timestamp* que indica el instante el momento en el que un usuario ha realizado una acción dentro de la plataforma de aprendizaje incluyendo fecha y hora de la acción.

La tabla de *logs* almacena las acciones realizadas en la plataforma tanto de los

estudiantes como de los profesores y de los administradores del sistema. Dado que la información necesaria para el estudio pertenece únicamente a los estudiantes es necesario realizar un filtrado. Este filtrado de las acciones involucra, además de la tabla de *logs*, otro conjunto de tablas de la plataforma que permiten identificar los estudiantes matriculados dentro de las asignaturas. La primera tabla es la correspondiente a las asignaturas (comúnmente `mdl_course`) que es la que contiene información relevante sobre la asignatura, en especial el identificador; en segundo lugar, es necesario seleccionar el rol de estudiante (`mdl_role_assignments`) y realizar la asociación entre los roles y las asignaturas (`mdl_context`), ya que un mismo usuario puede tener diferentes roles en función de la asignatura.

El conjunto de acciones utilizado en esta investigación corresponde a las acciones realizadas durante el curso académico 2017/2018 de la Universidad de Oviedo. Como se ha mencionado anteriormente, los datos de carácter personal fueron previamente anonimizados para cumplir con las leyes de protección de datos. Este proceso de anonimizado se ha realizado no solo sobre la tabla de logs sino sobre todo el conjunto de tablas de Moodle.

3.2. Filtrado de las asignaturas

El conjunto de registros disponibles almacena información de 5.112 asignaturas impartidas en la Universidad de Oviedo que utilizaron la plataforma de *e-learning* Moodle. Estas asignaturas pertenecen a diversas disciplinas como arte, humanidades, ciencias, ciencias sociales y jurídicas, e ingenierías, incluyendo tanto asignaturas de grados como de postgrados. El número de alumnos totales matriculados en las asignaturas es de 29.602. La plataforma de aprendizaje brindaba apoyo a estas asignaturas donde algunas eran totalmente online, otras semipresenciales y las últimas únicamente utilizaban la plataforma como un repositorio de contenido.

Una de las principales contribuciones de nuestra investigación es analizar la relación entre las acciones de los estudiantes en la plataforma con su rendimiento (Sección 1.2). Esta relación se analiza tanto a través de modelos predictivos como de modelos de agrupamiento. Debido a que no se dispone de la evaluación final de los alumnos en las asignaturas, el rendimiento académico de un alumno se ha definido como la evaluación obtenida a partir de las tareas evaluables en el LMS (para más detalle, véase la Sección 3.6). Por ello, es necesario hacer un filtrado de las asignaturas para únicamente utilizar aquellas que tengan al menos un recurso evaluable. Aquellas asignaturas que no disponen de este recurso evaluable no serán utilizadas en el análisis, debido a que no es posible relacionar acciones con una evaluación de los alumnos (no disponemos de las notas finales de todas las asignaturas).

Este filtrado de las asignaturas conlleva el uso de la información almacenada en las tablas `mdl_course`, `mdl_grade_items`, `mdl_grade_grades`, `mdl_context` y `mdl_role_assignments`. La tabla `mdl_course` permite identificar los cursos. Las tablas `mdl_grade_items` y `mdl_grade_grades` permiten reconocer los recursos evaluables en los cursos, filtrando solo aquellos que tengan una evaluación distin-

ta del valor vacío. Por último, las tablas `mdl_context` y `mdl_role_assignments` permiten identificar a los alumnos matriculados en la asignatura.

La Tabla 3.1 muestra un resumen de los datos de las asignaturas, estudiantes y número de acciones, tanto en el conjunto de datos original como en el conjunto de datos obtenido tras hacer el filtrado de las asignaturas.

Conjunto de datos original			**Conjunto de datos filtrado**		
Asignaturas	Alumnos	Registros (acciones)	Asignaturas	Alumnos	Registros (acciones)
5.112	29.602	47.097.824	699	15.944	8.540.418

Tabla 3.1: Comparativa entre el conjunto de datos original (izquierda) y el conjunto de datos filtrados (derecha)

3.3. Estimación de la duración de la asignatura

La plataforma Moodle utilizada en este estudio es capaz de proporcionar la duración de la asignatura a través de la información almacenada en la base de datos. Sin embargo, el proceso de anonimización realizado sobre los datos originales impide disponer de este dato. Esta falta de la información hace necesario definir un proceso para realizar una estimación de la duración de las asignaturas, ya que uno de los objetivos de la investigación consiste en realizar los análisis en instantes tempranos de la asignatura [13].

Dentro de la Universidad de Oviedo se engloban diferentes grados y postgrados de distintas disciplinas, lo que propicia la existencia de asignaturas de diferentes duraciones dentro de la misma plataforma. En el caso de los grados las asignaturas suelen tener una duración semestral o anual, mientras que en el caso de los postgrados la duración tiene una mayor variabilidad, incluso pudiendo haber asignaturas intensivas de una o dos semanas.

La información proporcionada en los registros permite conocer el comienzo de la asignatura, pero no su finalización, siendo necesario este valor para conocer la duración total. Con el objetivo de estimar esta fecha de finalización, tomamos 31 asignaturas semestrales (4,58 % del total) de las que se conocía su duración exacta para definir el mecanismo que permita estimar la duración de cualquier asignatura. Con este objetivo, las acciones de los estudiantes dentro de las asignaturas de control fueron ordenadas por la fecha de realización y se analizó el valor percentil de acciones que más se ajustase a la duración total del curso, comprobando que el percentil 95 % era el que mejores resultados ofrecía. Esto significa que, ignorando valores anómalos (*outliers*), el 5 % de las acciones tiene lugar una vez ha finalizado la asignatura. El cálculo de los valores anómalos se realizó mediante la Ecuación 3.1 [37]:

$$[Q_1 - 1{,}5 \times IQR, Q_3 + 1{,}5 \times IQR] \tag{3.1}$$

En la ecuación anterior Q_n indica el cuartil n, mientras que IQR indica el rango intercuartílico, es decir, $Q_3 - Q_1$.

3.4. Selección de los instantes de predicción

Realizar un análisis tras finalizar la asignatura, con todas las acciones de los alumnos almacenadas en los registros, permite estudiar y entender el proceso de aprendizaje. Sin embargo, en este instante ya no es posible realizar modificaciones tempranas en el proceso de aprendizaje de los alumnos. El principal objetivo de realizar un análisis temprano es realizar acciones correctivas o motivacionales sobre los alumnos, en función de la casuística en la que se encuentren. Es por todo ello por lo que esta investigación persigue realizar estos análisis en instantes tempranos de la asignatura.

No obstante, tratar de realizar análisis en instantes tempranos tiene como principal inconveniente que el número de acciones de los alumnos sea menor. Este déficit de acciones influye directamente en la precisión de los modelos, pues no se dispone de tanta información para detectar correctamente el comportamiento de los alumnos. Como hemos visto en el trabajo relacionado (Capítulo 2), esta precisión será mayor en instantes más avanzados de la asignatura, donde el número de acciones de los estudiantes es mayor.

Nuestra investigación persigue la construcción de modelos de predicción y de agrupamiento de alumnos en instantes tempranos de las asignaturas. Las variables de entrada construidas para esos modelos únicamente incluyen información de las acciones realizadas previamente al instante temprano seleccionado. Para ello, los instantes de predicción que hemos fijado son el 10 %, 25 %, 33 % y 50 % de la duración total de la asignatura. La selección de estos valores permite la realización de distintos modelos desde instantes tempranos en la impartición de la asignatura. Adicionalmente, también es posible analizar la tendencia en la precisión de los modelos a través de los distintos instantes de predicción teniendo en cuenta que se dispone de más información de los alumnos.

3.5. Generación de las variables independientes

La generación de variables independientes o predictoras (*features*) se define como un proceso que toma como entrada la información en crudo de los *logs* de Moodle y genera variables descriptivas del comportamiento de los alumnos que sirva como entrada a los modelos de aprendizaje automático supervisados y no supervisados.

Los datos almacenados por la plataforma pueden ser clasificados en información relacionada con las acciones realizadas por los estudiantes e información relacionada con su evaluación. Esta división permite la generación de variables independientes de dos tipologías diferentes. El primer grupo contiene aquellas acciones relativas al uso de recursos internos y externos (URL), uso de los foros, el acceso a la asignatura o a otros contenidos incluidos en ella (tareas y cuestionarios). Por otro lado, las variables relacionadas con la evaluación están relacionadas

con las calificaciones obtenidas por los estudiantes en diferentes tareas evaluables. Como se ha visto en el Capítulo 2, estas tipologías de variables están ampliamente documentadas en estudios similares.

La información almacenada por la plataforma y relacionada con las acciones registra cualquier interacción de los alumnos en las asignaturas. Dentro de este estudio se ha realizado una selección de acciones de los alumnos que son representativas dentro del conjunto de asignaturas. Estas acciones incluyen el acceso a contenidos de la asignatura, la realización de cuestionarios o la entrega de tareas, entre otras. Para filtrar y seleccionar la información de estas acciones es necesario hacer uso de los campos `module` y `action` de la tabla `mdl_log` descritos en la Sección 3.1. Por ejemplo, si el valor de la columna `module` es *course* y la columna `action` contiene el valor *view*, ese registro se corresponde con un acceso a una asignatura. Del mismo modo, si los valores son *resource* y *view* respectivamente, ese registro se corresponde con el acceso a un recurso de la asignatura. La Tabla 3.2 muestra las acciones seleccionadas para este estudio, identificando para cada una de ellas el valor almacenado en la base de datos de registros para ambas columnas (filtro para detectar el tipo de acción).

Descripción	`module`	`action`
Acceso a la asignatura.	*course*	*view*
Acceso a un recurso almacenado dentro de la plataforma (por ejemplo: PDF, Word, ...)	*resource*	*view*
Acceso a un recurso externo a la plataforma (URL a dirección externa).	*url*	*view*
Acceso y visualización de la descripción de una tarea.	*assign*	*view*
Envío del entregable de una tarea.	*assign*	*submit*
Acceso y visualización de la descripción de un cuestionario.	*quiz*	*view*
Comienzo de realización de un cuestionario.	*quiz*	*attempt*
Finalización y envío de un cuestionario.	*quiz*	*close attempt*
Acceso al foro de discusión.	*forum*	*view forum*
Acceso a una de las discusiones del foro.	*forum*	*view discussion*

Tabla 3.2: Descripción de las acciones seleccionadas para el estudio y sus valores de los campos `module` y `action`.

Al margen de las acciones como tal (tabla `mdl_logs`), para la generación de variables relacionadas con la evaluación también es necesario recuperar información de la tabla `mdl_grade_grades`. De esta tabla, únicamente se extrae aquella información relacionada con las tareas identificadas en las acciones de los alumnos.

Es importante remarcar que la construcción de las variables (acciones y evaluación) únicamente tiene en cuenta la información previa al instante en el que se realiza el análisis. Como se ha mencionado, esta tesis doctoral persigue la construcción de modelos tempranos de predicción y agrupamiento, utilizando la información únicamente disponible hasta el instante de predicción (véase la Sección 1.2). No obstante, es probable que en instantes de predicción muy iniciales

(10 % de la duración de la asignatura) algunas de las variables aún no tengan información suficiente para realizar la predicción con una precisión elevada.

Adicionalmente, dado que la naturaleza de las asignaturas es heterogénea, es necesario relativizar las variables de forma que puedan ser válidas para distintos tipos de asignaturas a la hora de realizar la predicción. Por ejemplo, todas las variables relacionadas con el tiempo hasta la realización de una acción están relativizadas respecto la duración de la asignatura.

El número total de variables generadas en el conjunto de datos (independientemente del instante de predicción) es de 63. De éstas, 55 corresponden a variables de tipo acción y 8 a variables relacionadas con la evaluación previa de los estudiantes (Anexo A y Anexo B).

3.5.1. Variables independientes tipo acción

Este tipo de variables se corresponde directamente con las acciones que realiza el estudiante dentro de la plataforma de aprendizaje. Como se ha mencionado en la sección anterior, las variables construidas están relativizadas, evitando el sesgo de los valores absolutos. Por ejemplo, la variable `CourseViewTime1` indica el primer acceso a la asignatura de forma porcentual relativa a la estimación de la duración de la asignatura que se ha realizado. Esta relativización permite que el significado de esa variable sea el mismo, independientemente de si la duración de la asignatura es una semana, un mes o seis meses. Por otro lado, otras variables como por ejemplo `ResourceViewPct` muestran el porcentaje de accesos a los recursos internos por parte de un alumno con respecto al número total de accesos a los recursos internos por parte de todos los alumnos.

Para más información, el Anexo A muestra el listado completo de variables de esta tipología, así como su descripción.

3.5.2. Variables independientes tipo evaluación

Este tipo de variables utiliza, además de la información de las acciones, la evaluación como tal de las tareas entregadas por los alumnos en la plataforma. Esta información no está almacenada directamente en los registros de las acciones, sino en una tabla específica para las evaluaciones (`mdl_grade_grades`).

Dentro de las asignaturas es común ofrecer a los alumnos ciertas tareas como opcionales. Las tareas opcionales influyen la evaluación final de los alumnos, pero en muchos de los casos su peso es menor que el de las tareas obligatorias. Debido a esta distinta naturaleza, es interesante generar variables de evaluación teniendo en cuenta la obligatoriedad u opcionalidad de las tareas y analizar su impacto en los resultados. Sin embargo, Moodle no proporciona indicadores que permitan identificar si una tarea es opcional o no.

Para solventar esta falta de información se utilizó un enfoque similar al descrito en la Sección 3.3. Para ello se utilizó un conjunto de 30 asignaturas que ofrecían tanto tareas opcionales como obligatorias. Para cada una de las 30 asignaturas, se computó el porcentaje de alumnos que entregaba cada tarea para establecer un

umbral que permitiese distinguir las tareas opcionales de las obligatorias. Analizando los porcentajes obtenidos, se comprobó que todas las tareas obligatorias eran entregadas por al menos el 40 % de los alumnos.

Este umbral estimado nos permite la generación de variables tanto para evaluaciones de tareas obligatorias como para opcionales, enriqueciendo así la información introducida al modelo. Por ejemplo, la variable `AccomplishOptionalPctGraded` proporciona el porcentaje de tareas opcionales entregadas por el alumno hasta el instante de análisis. Por otro lado, la variable `AccomplishMandatoryPercentileGrade` indica el percentil en el que se sitúa la evaluación de las tareas obligatorias de un alumno en una asignatura con respecto al resto de alumnos.

Para más información, el Anexo B muestra el listado completo de variables de esta tipología, así como su descripción.

3.6. Generación de la variable dependiente o *target*

Este estudio persigue la creación de modelos supervisados y no supervisados en instantes tempranos de una asignatura. En el caso de los modelos supervisados, es necesario disponer de una variable dependiente o *target* a predecir, independientemente de si lo que se quiere predecir es un valor numérico (modelos de regresión) o una categoría (modelos de clasificación).

La variable dependiente es el rendimiento o evaluación de un alumno en una asignatura. Sin embargo, debido al proceso de anonimización realizado sobre la base de datos, la evaluación final del alumno en la asignatura no está disponible. Por tanto, para la construcción de la variable dependiente se tendrán en cuenta las evaluaciones del alumno dentro de la plataforma de forma agregada, puesto que esta información sí está disponible.

En lugar de definir una única forma de agregación de las evaluaciones de los estudiantes, se han definido varias ecuaciones para seleccionar aquella que proporcione valores más próximos a la evaluación real. Para ello se utilizaron 30 asignaturas de control donde las evaluaciones finales estaban disponibles (mismo conjunto que en la sección anterior).

$$10 \times \left(\alpha \frac{\sum \textit{evaluación tareas obligatorias}}{\textit{entrega tareas obligatorias}} + (1-\alpha) \frac{\sum \textit{evaluación tareas opcionales}}{\textit{entrega tareas opcionales}} \right) \quad (3.2)$$

$$10 \times \left(\alpha \frac{\sum \textit{evaluación tareas obligatorias}}{\textit{tareas obligatorias}} + (1-\alpha) \frac{\sum \textit{evaluación tareas opcionales}}{\textit{tareas opcionales}} \right) \quad (3.3)$$

Definimos dos ecuaciones de estimación de la evaluación de estudiantes (Ecuación 3.2 y Ecuación 3.3) dependientes del parámetro α. Ambas ecuaciones utilizan como numeradores los sumatorios de las evaluaciones de las tareas obligatorias y

opcionales del alumno en la asignatura. Los denominadores de la primera ecuación consideran solo las entregas (tareas presentadas), mientras que la segunda ecuación tiene en cuenta el número total de tareas. En esta segunda ecuación, la no entrega de una tarea está considerada como un 0.

Las ecuaciones utilizan un coeficiente α para otorgar distintos pesos a las tareas obligatorias y opcionales. El objetivo es tratar de ajustar de la forma más correcta su valor de forma que se minimice el error en la estimación de la evaluación. El peso α toma un valor comprendido dentro de los valores [0, 1], teniendo en cuenta únicamente las tareas obligatorias cuando α=1.

Para ver qué ecuación y valor α aproximan más la nota final a la estimada, utilizamos las evaluaciones de los alumnos en las 30 asignaturas de control. Para distintos valores de α calculamos el error cometido en las estimaciones (Tabla 3.3) usando las métricas error cuadrático medio (*mean square error* o MSE), raíz del error cuadrático medio (*root mean square error* o RMSE), error medio absoluto (*mean absolute error* o MAE), el coeficiente de determinación (R^2) y el R^2 ajustado [38]. Adicionalmente también se incluyen los coeficientes de correlación de Pearson y Spearman.

Ecuación	α	**MSE**	**RMSE**	**MAE**	**R^2**	**R^2 ajustado**	**Pearson**	**Spearman**
	1	0,0070	0,0881	0,0756	0,3439	0,3146	0,6110	0,6272
	0,75	0,0064	0,0810	0,0694	0,3742	0,3424	0,6650	0,6826
(3.2)	0,5	0,0063	0,0793	0,0680	0,3820	0,3495	0,6789	0,6969
	0,25	0,0068	0,0861	0,0738	0,3519	0,3219	0,6253	0,6419
	0	0,0077	0,0967	0,0830	0,3132	0,2865	0,5565	0,5713
	1	0,0062	0,0841	0,0710	0,4111	0,4043	0,7165	0,7798
	0,75	0,0055	0,0755	0,0638	0,4579	0,4503	0,7981	0,8685
(3.3)	**0,5**	**0,0054**	**0,0733**	**0,0619**	**0,4716**	**0,4638**	**0,8218**	**0,8944**
	0,25	0,0059	0,0798	0,0674	0,4334	0,4262	0,7553	0,8220
	0	0,0068	0,0922	0,0778	0,3751	0,3689	0,6537	0,7114

Tabla 3.3: Errores obtenidos en la estimación de la evaluación de los alumnos (en negrita el mejor resultado por tener el menor error).

La Tabla 3.3 muestra los resultados obtenidos para ambas ecuaciones y los distintos valores de α. Para interpretar correctamente los valores de los errores MSE, RMSE y MAE es importante tener en cuenta que las evaluaciones están escaladas al intervalo 0-1. El menor error se obtiene para la Ecuación 3.3 con α=0,5, valor con el que se explica el 47,16 % de la varianza (métrica R^2). Nótese que MSE, RMSE y MAE miden errores y por tanto se buscan valores más próximos a 0; R^2, R^2 ajustado, Pearson y Spearman muestran correlaciones por lo que a valores más elevados mejor resultado.

Es importante destacar que ningún modelo se crea con todos los valores de las variables utilizadas para calcular el rendimiento del alumno. Es decir, en ninguno de los modelos el instante de predicción incluye todas las evaluaciones existentes para calcular el rendimiento de los alumnos. Por ello, la medición del rendimiento de los modelos se realiza sin que éstos hayan sido entrenados con la información empleada para calcular el valor de la variable dependiente, evaluando así realmente su capacidad predictiva.

Capítulo 4

Modelos predictivos del rendimiento

Tras la generación del conjunto de datos, tal y como se ha descrito en el capítulo anterior, el siguiente paso es mostrar la primera contribución de esta tesis doctoral (Sección 1.2): predicción del rendimiento académico de los alumnos en las tareas de una plataforma de aprendizaje, utilizando únicamente la información de actividad generada hasta distintos instantes de tiempo tempranos del curso. Para ello, construiremos modelos predictivos a partir del *dataset* creado según se ha explicado en la sección anterior. Posteriormente, también se analizará la influencia de cada una de las variables en la predicción del rendimiento (segunda contribución del estudio).

4.1. Variable dependiente del modelo predictivo

El capítulo anterior describe cómo generar la variable dependiente que representa la estimación de la evaluación de los alumnos en las asignaturas. Sin embargo, esta variable es continua, con una escala [0, 10]. Puesto que el objetivo de la investigación es identificar alumnos en riesgo alto de no superar la asignatura, definimos una nota de corte para convertir un problema de regresión en un problema de clasificación, más acorde a los objetivos de este estudio. Adicionalmente, también definimos notas de corte para detectar alumnos excelentes y el umbral entre aprobado y suspenso.

La primera nota seleccionada para la discretización de la variable en dos grupos excluyentes es 2,5. De esta forma, se genera una variable dicotómica que permite identificar aquellos alumnos en claro riesgo de no superar la asignatura (evaluación $\leq$ 2,5). También se utiliza la nota de corte 8,5 para detectar alumnos excelentes (evaluación $\geq$ 8,5). Por último, con el objetivo de analizar el escenario de aprobado / suspenso, la variable rendimiento también es discretizada utilizando la nota 5,0.

La Tabla 4.1 muestra el número de estudiantes en función de las notas de corte definidas. Se aprecia cómo será más difícil identificar a los alumnos que aprueban o suspenden la asignatura que los casos extremos, ya que el grupo de alumnos

aprobados / suspensos (39 % / 61 %) está más equilibrado que en los otros casos (26 % / 74 % y 20 % / 80 %, para las notas 2,5 y 8,5 respectivamente).

Nota de corte	$\leq$ **2,5**	$<$ **5**	$\geq$ **8,5**
Porcentaje de alumnos en el *dataset*	25,76 %	39,39 %	20,34 %

Tabla 4.1: Porcentaje de alumnos en el conjunto de datos para cada una de las notas de corte.

La definición de tres notas de corte implica la generación de tres modelos distintos utilizando las mismas variables independientes, pero con diferentes variables dependientes (todas ellas dicotómicas). Multiplicando estos tres modelos por el número de instantes de predicción (10 %, 25 %, 33 % y 50 % de la duración de la asignatura) resulta un total de doce modelos diferentes a entrenar y evaluar.

4.2. Metodología de generación de modelos predictivos

El siguiente paso es la generación de los modelos de predicción del rendimiento de los estudiantes en instantes tempranos. Estos modelos permitirán la predicción de la superación o no de las notas de corte seleccionadas (alumnos en riesgo, excelentes y aprobados).

La generación de los modelos, independientemente del instante de predicción, sigue una misma metodología. En la Figura 4.1 se muestran los pasos seguidos para la obtención los modelos. Los nodos azules muestran la estrategia seguida, mientras que los naranjas indican alternativas o técnicas probadas durante la investigación pero que fueron descartadas posteriormente por no ofrecer resultados mejores.

Como muestra la Figura 4.1, se han utilizado varias técnicas para el equilibrado de los datos y para la discretización de las variables, con el objetivo obtener los modelos con mayor rendimiento. Debido a que el conjunto de datos no está equilibrado (Tabla 4.1) se probaron las técnicas *Random Under Sampling*, *Tomek-Links* [39], *Random Over Sampling* y *SMOTE* [40] para equilibrar el conjunto de datos [41]. En el caso de la discretización, las variables se discretizaron usando las técnicas de *equal width*, *equal frecuency* y *ChiMerge* con tres, cuatro y cinco intervalos [42]. Todas estas técnicas fueron descartadas en la metodología final ya que no proporcionaban modelos con mejores resultados.

A partir del conjunto de datos original se han generado dos subconjuntos para entrenamiento y evaluación (*test*). El primero está formado por el 80 % de los registros, mientras que el segundo utiliza el 20 % restante. Para la realización de esta división se ha utilizado un método de muestreo aleatorio estratificado [43]. Este proceso se ha realizado 30 veces, manteniendo los mismos porcentajes en la división. El método de división del conjunto de datos permite mantener la misma proporción de registros para cada clase que en el conjunto original. Con los resultados del conjunto de 30 iteraciones se ha medido la media, desviación estándar y el intervalo de confianza del 95 % de las métricas exactitud (*accuracy*), F_1-*score*

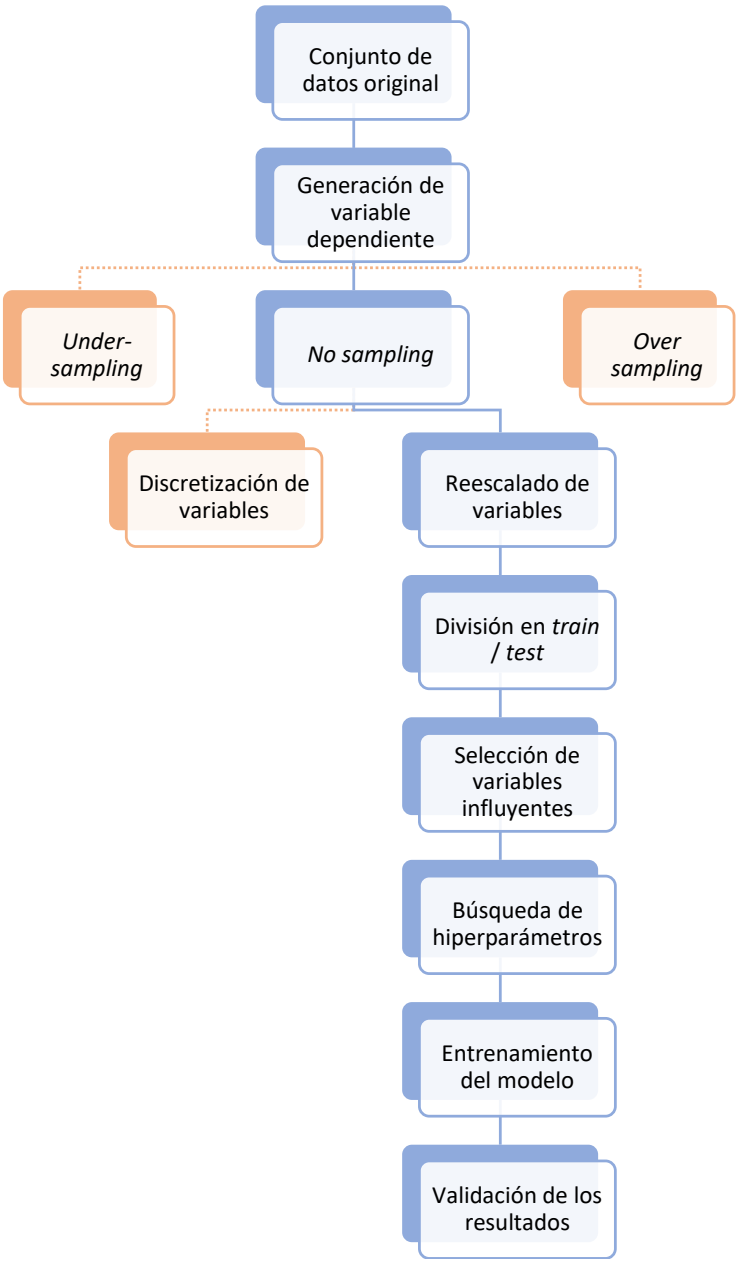

Figura 4.1: Etapas de la metodología seguidas para la construcción de modelos predictivos.

y área bajo la curva ROC (AUC) [44]. Estas mediciones permiten comparar los modelos obtenidos entre sí y analizar si existen diferencias significativas entre ellos.

Para la generación de los 12 modelos de clasificación binaria mencionados con anterioridad, se han utilizado los siguientes algoritmos (entre paréntesis la abreviatura utilizada más adelante): árboles de decisión CART (DT), Naïve Bayes (NB), Regresión Logística (LR), *Multilayer Perceptron* (MLP) y máquinas de vector soporte (SVM). La implementación utilizada de estos algoritmos es la proporcionada por el framework `scikit-learn` en su versión 0.22.2. Atendiendo a esta implementación, los nombres de las clases para esos algoritmos son `DecisonTreeClassifier` (DT), `GaussianNB` (NB), `LogisticRegression` (LR),

`MLPClassifier` (MLP) y `SVC` (SVM). El código se ha implementado usando Python 3.7.6.

Dado que algunos de los algoritmos empleados (MLP y SVM) ofrecen mejores resultados con variables normalizadas [45], todas las variables han sido normalizadas a una escala entre 0 y 1. Tras este proceso, se realiza una selección de las variables más importantes para el modelo, así como una búsqueda de los mejores hiperparámetros. El algoritmo utilizado para la selección de las variables es `RFECV`, incluido dentro de `scikit-learn`, que permite realizar una eliminación recursiva de variables [46]. El mejor número de variables es seleccionado con validación cruzada aleatoria con 3 iteraciones (`StratifiedKFold` en Python), usando como métrica de evaluación la exactitud (*accuracy*). Adicionalmente, también se añade la condición de que deben existir al menos diez variables seleccionadas.

En el caso de la búsqueda de los mejores hiperparámetros el enfoque es similar. Se parten de los parámetros base proporcionados por la implementación de los algoritmos para posteriormente realizar una búsqueda exhaustiva en paralelo entre un conjunto de parámetros definidos (`RandomizedSearchCV` en Python). Al igual que en el caso anterior, también se utiliza la validación cruzada aleatoria con tres iteraciones (`StratifiedKFold` en Python), usando como métrica de evaluación la exactitud. El Anexo C muestra tanto los hiperparámetros configurados para la búsqueda como el resultado de la misma para cada algoritmo en cada instante de tiempo y para cada nota de corte.

La ejecución de los procesos de selección de variables, selección de hiperparámetros y entrenamiento de los modelos se ha ejecutado de forma paralela dentro de un único servidor. Este es un Dell PowerEdge R530 con 2 microprocesadores Intel Xeon E5-2620 v4 a 2.1 GHz con 128GB DDR4 a 2400MHz de RAM. El sistema operativo instalado es un CentOS versión 7.4-1708 para máquinas de 64 bits.

4.3. Resultados de los modelos predictivos

Una métrica muy común para medir la bondad de los modelos de clasificación es la exactitud (*accuracy*)[1]. Sin embargo, existen situaciones donde otras métricas de bondad proporcionan información valiosa para explicar los resultados de una forma más completa. En esta investigación, hemos utilizado la exactitud, distintas medidas F_β [26] y el área bajo la curva ROC [20] como métricas para medir la bondad de los modelos y comparar su rendimiento.

La Figura 4.2 muestra la exactitud obtenida por los distintos modelos con cada una de las notas de corte y los instantes de predicción definidos. Adicionalmente se incluye dentro de la figura una línea discontinua que muestra la clase mayoritaria del conjunto de datos utilizado. Esta figura permite comprobar la evolución de la exactitud según avanza la asignatura, para cada una de las notas de corte seleccionadas.

La Tabla 4.2 muestra los valores AUC obtenidos para los cinco algoritmos

[1] *Accuracy* también puede traducirse como precisión, pero podría causar confusión con otra métrica llamada *precision*, por lo que es común usar exactitud.

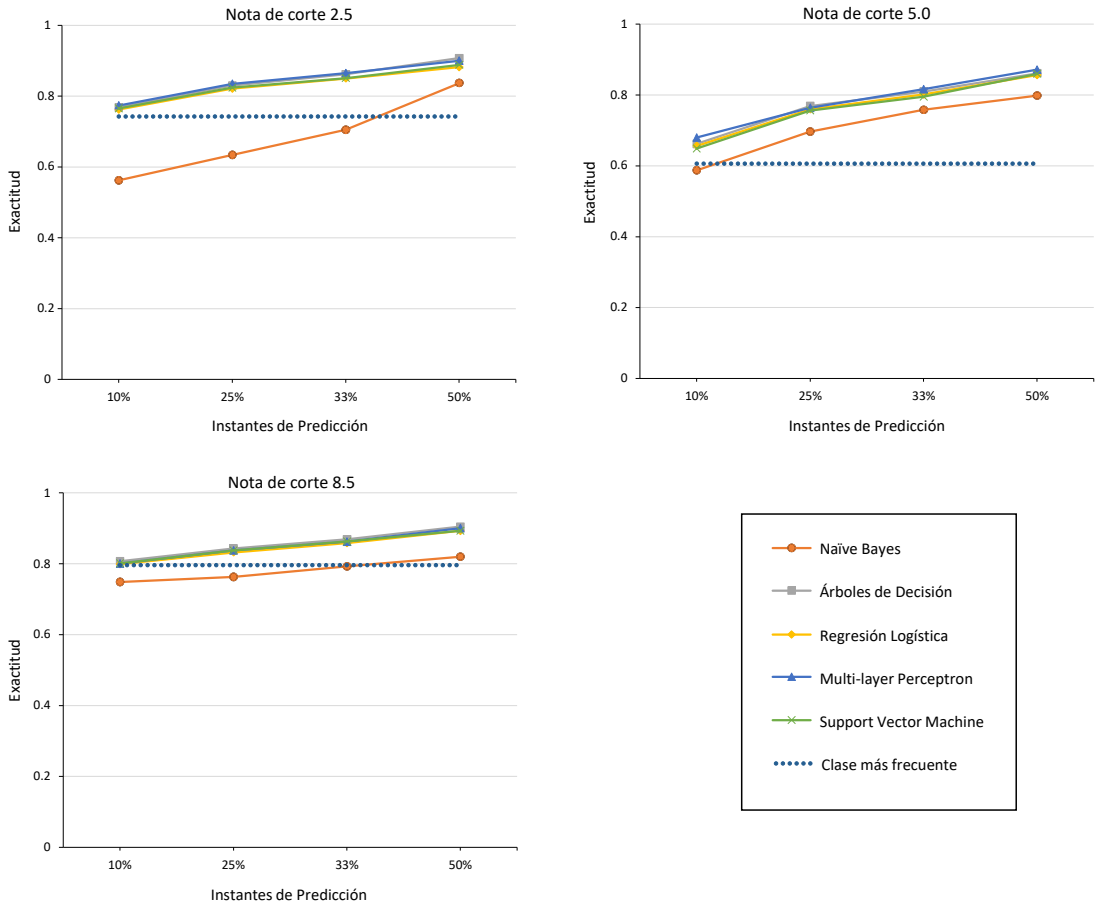

Figura 4.2: Evolución de la exactitud de los modelos para las distintas notas de corte e instantes de predicción.

utilizados. El AUC mide el área bajo la curva ROC teniendo en consideración el diagnóstico de las tasas de verdaderos y falsos positivos de un clasificador binario [47]. Dentro de la tabla están resaltados, en negrita, los mayores valores para cada uno de los modelos con un mismo instante de predicción y nota de corte. En caso de que no existan diferencias significativas entre los resultados de dos algoritmos (los intervalos de confianza del 95 % se solapan o valor p (*p-value*) $< 0{,}05$ para la prueba t de *Student*), ambos valores aparecerán en negrita.

Los resultados mostrados en la Figura 4.2 y en la Tabla 4.2 muestran que el algoritmo que mejores resultados ofrece es el *MultiLayer Perceptron* (MLP). No obstante, existen situaciones donde el resultado no es significativamente mejor que otros algoritmos como los árboles de decisión (DT).

La Figura 4.3 muestra las curvas ROC para los modelos generados con el algoritmo MLP, en cada una de las notas de corte e instantes de tiempo. Esta figura permite apreciar que el rendimiento de los modelos se incrementa conforme crece el instante de predicción, para las tres notas de corte seleccionadas.

La Tabla 4.3 muestra la comparativa entre la exactitud obtenida en los modelos obtenidos con el algoritmo MLP y la clase mayoritaria del dataset. La comparación de ambos valores no da información adicional sobre la capacidad predictiva de los modelos, al no estar los conjuntos de datos equilibrados (véase Tabla 4.1).

Momento de predicción	Nota de corte 2,5	Nota de corte 5,0	Nota de corte 8,5	Algoritmo
10 %	0,70501	0,67395	0,60651	NB
	0,73664	0,68949	0,67235	DT
	0,72470	0,71220	0,65633	LR
	0,76016	**0,72783**	**0,68783**	MLP
	0,70990	0,71260	0,63956	SVC
25 %	0,79181	0,74912	0,70559	NB
	0,84986	0,82854	**0,82014**	DT
	0,84773	0,83737	0,78587	LR
	0,87717	**0,84596**	**0,82083**	MLP
	0,85885	0,83609	0,79264	SVC
33 %	0,81336	0,81923	0,75494	NB
	0,89871	0,88324	**0,88024**	DT
	0,88105	0,88386	0,83148	LR
	0,91724	**0,89223**	**0,88046**	MLP
	0,89199	0,88375	0,83206	SVC
50 %	0,87335	0,88090	0,80257	NB
	0,93633	0,93506	**0,93515**	DT
	0,92493	0,93435	0,87315	LR
	0,95834	**0,94669**	**0,93199**	MLP
	0,93858	0,93388	0,89550	SVC

Tabla 4.2: Valores AUC para cada uno de los modelos obtenidos por nota de corte e instante de tiempo. En negrita los mejores valores obtenidos para cada instante de predicción y nota de corte.

Por otro lado, también se ha realizado un análisis de distintos valores F_β, incluyendo F_1, $F_{0,5}$ y F_2 (Tabla 4.4). Este enfoque nos permitirá analizar el coste de los falsos positivos y los falsos negativos (errores tipo I y tipo II) a la hora de predecir el rendimiento de los alumnos.

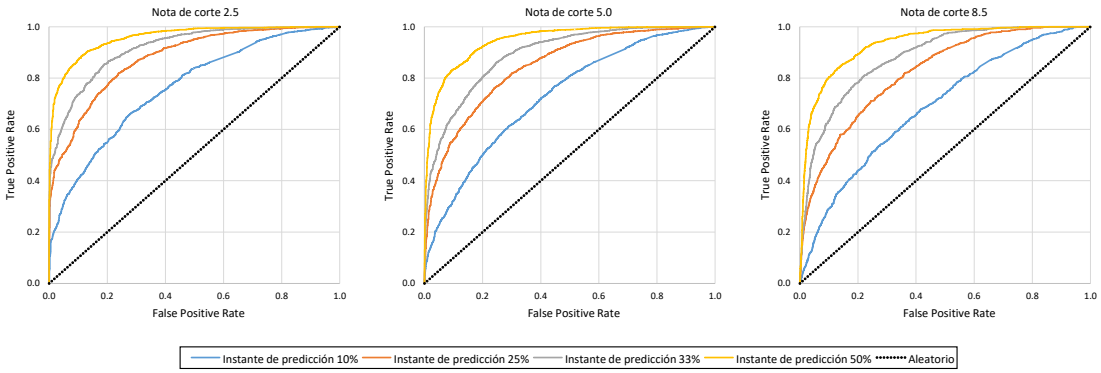

Figura 4.3: Curvas ROC obtenidas con el algoritmo MLP en las distintas notas de corte e instantes de predicción.

Nota de corte	Algoritmo	Instante de predicción 10 %	25 %	33 %	50 %
2,5	Clase Mayoritaria	74,24 %	74,24 %	74,24 %	74,24 %
	MLP	77,38 %	83,49 %	86,50 %	90,02 %
5	Clase Mayoritaria	60,61 %	60,61 %	60,61 %	60,61 %
	MLP	67,99 %	76,37 %	81,61 %	87,17 %
8,5	Clase Mayoritaria	79,66 %	79,66 %	79,66 %	79,66 %
	MLP	80,08 %	83,75 %	86,24 %	90,06 %

Tabla 4.3: Exactitud del algoritmo MLP comparado con el porcentaje de la clase mayoritaria del dataset.

Nota de corte	F_β	Instante de predicción 10 %	25 %	33 %	50 %
2,5	F_1	MLP (0,8634)	MLP (0,8944)	MLP (0,9121)	DT (0,9382)
	$F_{0,5}$	MLP (0,8130)	MLP (0,8681)	DT (0,8970)	DT (0,9300)
5,0	F_1	MLP (0,7527)	MLP (0,8129)	MLP (0,8525)	MLP (0,8941)
	$F_{0,5}$	MLP (0,7133)	DT (0,7948)	DT (0,8382)	MLP (0,8781)
8,5	F_1	DT (0,8909)	DT (0,9074)	DT (0,9217)	DT (0,9419)
	F_2	DT (0,9370)	DT (0,9457)	DT (0,9464)	DT (0,9540)

Tabla 4.4: Mejores resultados en las métricas $F_{0,5}$, F_1 y F_2 para las distintas notas de corte e instantes de predicción.

4.4. Discusión de los resultados de los modelos predictivos

Analizando la Figura 4.2 y la Tabla 4.2 es posible comprobar cómo a medida que la asignatura avanza, las predicciones de los modelos son más precisas. Este comportamiento es común en todas las notas de corte seleccionadas. El principal motivo para esta mejora es que los registros proporcionan más información o más acciones realizadas por los estudiantes a medida que el instante de tiempo avanza, lo que posibilita construir modelos más precisos. Esto nos muestra cómo no se deben construir modelos predictivos tempranos con toda la información de las asignaturas, puesto que los parámetros de los modelos cambian en función del instante de predicción [13].

Con la excepción del algoritmo Naïve Bayes, los algoritmos muestran un comportamiento en las predicciones bastante similar, mostrando diferencias en su exactitud inferiores al 5 %. Adicionalmente, los modelos obtenidos siempre superan el rendimiento de la clasificación basada en la clase mayoritaria.

El algoritmo que mejores resultados ofrece, en líneas generales, es el *MultiLayer Perceptron* (MLP). En el instante de análisis 10 %, este algoritmo es capaz de clasificar correctamente el 67,99 % de los alumnos que van a superar o no la asig-

natura (nota de corte 5,0). Este valor es un 7,38 % mayor que la clase mayoritaria del conjunto de datos (véase Tabla 4.3). Esta diferencia con la clase mayoritaria decrece hasta un 3,14 % en el caso de la detección de alumnos en riesgo (nota de corte 2,5) y hasta un 0,42 % en la clasificación de alumnos excelentes (nota de corte 8,5), siempre en el instante 10 % de la asignatura (predicción muy temprana). La diferencia entre la clase mayoritaria y la exactitud de los modelos es menor en estos dos últimos umbrales debido a que hay menor margen de mejora, al estar el conjunto de datos mucho menos equilibrado.

Como se ha mencionado, la exactitud de los modelos va en aumento a medida que el instante de predicción avanza. En el último instante seleccionado (50 % de la duración de la asignatura), se consigue una exactitud superior al 90 %, tanto para la detección de alumnos en riesgo como excelentes. Para el caso de la detección de los alumnos aprobados, el valor obtenido para esta métrica en el instante 50 % es del 87,2 %.

La Tabla 4.2 muestra las diferencias entre algoritmos utilizando la métrica AUC. En todos los casos, el algoritmo MLP es el que mejores valores ofrece, existiendo tres casos donde no existe diferencia significativa con los árboles de decisión CART (DT). En los tres casos donde no existen diferencias significativas se trata de detectar los alumnos excelentes. En la Tabla 4.2 también es posible comprobar cómo en instantes de tiempo más avanzados los valores del AUC son mayores, tal y como también se muestra en la Figura 4.3. Este comportamiento sigue el mismo patrón que la exactitud discutida con anterioridad.

Una discusión importante de nuestro trabajo es considerar el coste de los falsos positivos y falsos negativos. Estos tipos de errores en los modelos pueden acarrear situaciones tales como que un alumno no sea identificado como en potencial riesgo de no superar la asignatura y que, por tanto, no sea posible adaptar su proceso de aprendizaje para revertir la situación real. Para ello es posible utilizar diferentes medidas F_β, otorgando diferentes pesos al parámetro β según la Ecuación 4.1.

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{(\beta^2 \times precision) + recall} \tag{4.1}$$

En este análisis se considera positivo a un estudiante que supera cada una de las notas seleccionadas (2,5, 5,0 o 8,5). Para los modelos con umbrales 2,5 y 5,0 se persigue obtener modelos con alta precisión (*precision*), reduciendo el número de alumnos en riesgo o alumnos que no superan la asignatura mal clasificados (reducir falsos positivos). Esto significa que el coste de clasificar incorrectamente a alumnos en riesgo o suspensos es mayor que la clasificación errónea de alumnos sin riesgo o que superan la asignatura (se le aplicarían medidas correctivas para mejorar su aprendizaje, aunque no fuesen realmente necesarias). Para este tipo de escenarios utilizamos la medida $F_{0,5}$ ($\beta = 0{,}5$) que otorga el doble de peso a la precisión (*precision*) frente a la exhaustividad (*recall*) [48].

En el caso de alumnos excelentes (superar el umbral 8,5) la casuística es la contraria. Se persigue identificar correctamente a aquellos alumnos excelentes para poder potenciar su aprendizaje mediante otras tareas opcionales. Por ello,

lo que se persigue es reducir el número de alumnos excelentes no detectados por los modelos (falsos negativos). Para este enfoque, la medida F_2 ($\beta = 2$) da el doble de peso al *recall* que a la *precision*.

La Tabla 4.4 muestra los mejores valores para las métricas $F_{0,5}$ y F_1 para los umbrales de corte 2,5 y 5,0 y las métricas F_1 y F_2 para el umbral de corte 8,5. Los mejores resultados se obtienen tanto en el algoritmo MLP como en los árboles de decisión, produciéndose los mismos resultados que con la métrica AUC. Esto demuestra que tanto el algoritmo MLP como el DT son los dos más adecuados para clasificar el rendimiento de los estudiantes.

Tras los análisis realizados, podemos afirmar que hemos alcanzado la **Contribución 1** de esta tesis doctoral: es posible predecir el rendimiento académico de los alumnos en las tareas desarrolladas en las plataformas de aprendizaje, utilizando únicamente las acciones realizadas en ellas. Esta predicción puede ser realizada tanto en instantes muy tempranos de la asignatura (10 % de la duración) hasta instantes más avanzados (50 % de la duración) de la misma. En el instante 10 %, las diferencias de rendimiento respecto a la clase mayoritaria son significativas para aprobados/suspensos, pero no para la detección de alumnos excelentes y en riesgo elevado de suspender. A medida que los modelos disponen de más información de las acciones de los alumnos, las predicciones son más precisas, obteniendo diferencias significativas para todos los casos en el 25 % de la duración de la asignatura. Los algoritmos con mejores resultados han sido MultiLayer Perceptron (MLP) y árboles de decisión (DT).

4.5. Análisis de las variables independientes

Una de las características más buscadas de los modelos es su capacidad predictiva sobre nuevas observaciones. Existen situaciones donde, además de esta capacidad predictiva, es necesario analizar qué variables independientes influyen más en la predicción. La segunda contribución de este estudio está centrada en analizar qué acciones de los estudiantes dentro de la plataforma de aprendizaje influyen más en su rendimiento académico.

No todos los algoritmos utilizados permiten realizar un análisis de caja blanca que posibilite analizar la influencia de cada variable en la predicción (no todos los modelos poseen explicabilidad). Uno de estos modelos son las redes neuronales, incluyendo la arquitectura multicapa utilizado en nuestro estudio (*MultiLayer Perceptron*). Dado que los resultados mostrados en la Tabla 4.2, la Tabla 4.3 y la Figura 4.2 muestran que los árboles de decisión tienen una capacidad predictiva similar al MLP, utilizamos este algoritmo para analizar la influencia de las variables, puesto que permite realizar el análisis de caja blanca. Los árboles de decisión describen caminos o rutas desde el nodo raíz hasta los nodos hoja que pueden ser interpretados como reglas de clasificación, donde cada condición es una comparación entre una variable y un valor [49].

Cuando se genera en árbol de decisión, la ganancia de información se usa para determinar qué variable independiente proporciona la mayor información sobre una de las clases a predecir. Una de las medidas de ganancia de información es la

importancia o índice Gini, definida como la suma del número de divisiones que incluyen esa variable, en relación con el número de instancias o individuos que divide [50, 51].

La Tabla 4.5 muestra las cinco variables con mayor importancia Gini para cada uno de los modelos generados. El 56,67 % de las variables con más importancia Gini pertenecen al grupo de las variables relacionadas con la evaluación (Anexo B), mientras que el 43,33 % de ellas pertenecen a las variables de las acciones de los alumnos (Anexo A). Para el instante de predicción 10 %, los modelos utilizan menos variables de evaluación porque es menos común tener tareas evaluables en las primeras etapas de la asignatura. El peso de las variables de evaluación crece a medida que aumenta el momento de la predicción. Ese peso es del 43 %, 56 %, 62 % y 64 % para, respectivamente, los instantes de predicción 10 %, 25 %, 33 % y 50 %.

La variable independiente con mayor influencia en la predicción es `AccomplishMandatoryGrade`. Esta variable considera las evaluaciones obtenidas por el alumno previamente al instante de predicción, considerando las tareas no entregadas como un cero. Esta variable es la más importante en todos los modelos obtenidos a excepción de uno de ellos (umbral 2,5 e instante de predicción 50 %) donde ocupa el segundo lugar. En general, las variables relacionadas con la evaluación tienen más influencia en la predicción según avanza la asignatura. Esto es debido a que en los instantes iniciales es posible que los alumnos no tengan ninguna tarea evaluable realizada.

La Tabla 4.5 muestra cómo el acceso a los recursos internos proporcionados por los profesores (`ResourceViewPct` y `ResourceViewUniquePct`) son variables influyentes para detectar alumnos en riesgo y aquéllos que aprueban/suspenden (umbrales 2,5 y 5,0). Sin embargo, estas variables no son relevantes para la detección de los alumnos excelentes dentro de la asignatura.

La variable `CourseViewPct` mide el porcentaje de acceso a la asignatura con respecto al total de accesos de todos los estudiantes. Esta variable es influyente únicamente para la detección de los alumnos excelentes. Esto indica que los estudiantes excelentes acceden más frecuentemente a la plataforma de aprendizaje, siendo más influyente este parámetro en etapas iniciales de la asignatura.

Las variables relacionadas con la entrega de tareas tienen cierta influencia a la hora de detectar los alumnos en riesgo o aprobados/suspensos. Sin embargo, esta influencia no se ve reflejada a la hora de clasificar a los alumnos excelentes. De la misma forma, las variables relacionadas con los cuestionarios no tienen una gran correlación con la predicción del rendimiento, ya que únicamente aparecen en dos de los doce modelos obtenidos.

Por último, es importante indicar que las variables relacionadas con los foros de discusión (`ForumViewForumPct` y `ForumViewDiscussionPct`) no aparecen entre las variables influyentes para ninguno de los modelos obtenidos. Esto parece indicar que la interacción y comunicación a través de este mecanismo no guarda relación con el rendimiento de los alumnos.

Este análisis conforma la **Contribución 2** centrada en identificar las variables

	Instante de predicción			
	10 %	**25 %**	**33 %**	**50 %**
Nota de corte 2,5	AccomplishMandatory-Grade (0,36) ResourceViewUniquePct (0,12) AssignSubmitUniquePct (0,07) AccomplishOptionalPct-Graded (0,05) AccomplishOptional-PercentileGrade (0,04)	AccomplishMandatory-Grade (0,66) ResourceViewUniquePct (0,08) AssignSubmitUniquePct (0,04) AssignViewUniquePct (0,02) AccomplishMandatory (0,02)	AccomplishMandatory-Grade (0,29) AccomplishMandatory-PctGraded (0,27) AccomplishMandatory-PercentileGrade (0,11) AccomplishMandatory (0,06) AssignViewUniquePct (0,04)	AccomplishMandatory-PctGraded (0,38) AccomplishMandatory-Grade (0,35) AccomplishMandatory (0,09) ResourceViewUniquePct (0,02) AssignViewUniquePct (0,02)
Nota de corte 5,0	AccomplishMandatory-Grade (0,17) QuizCloseAttempt-UniquePct (0,12) AssignSubmitPct (0,07) ResourceViewPct (0,06) AccomplishMandatory-PctGraded (0,06)	AccomplishMandatory-Grade (0,56) AccomplishMandatory-PercentileGrade (0,17) AssignViewUniquePct (0,03) ResourceViewUniquePct (0,03) AssignSubmitUniquePct (0,02)	AccomplishMandatory-Grade (0,72) AssignViewUniquePct (0,05) AssignSubmitUniquePct (0,03) AccomplishMandatory-PctGraded (0,02) ResourceViewUniquePct (0,02)	AccomplishMandatory-Grade (0,77) AccomplishMandatory-PercentileGrade (0,09) AssignViewUniquePct (0,02) AccomplishMandatory (0,02) AssignSubmitUniquePct (0,01)
Nota de corte 8,5	AccomplishMandatory-Grade (0,42) CourseViewPct (0,18) AccomplishMandatory-PercentileGrade (0,13) AccomplishOptional-PctGraded (0,09) AccomplishMandatory-PctGraded (0,07)	AccomplishMandatory-Grade (0,76) AccomplishMandatory-PctGraded (0,07) CourseViewPct (0,06) AccomplishMandatory-PercentileGrade (0,05) AssignViewPct (0,01)	AccomplishMandatory-Grade (0,8) AccomplishMandatory (0,08) CourseViewPct (0,04) QuizAttemptPct (0,01) AssignViewPct (0,01)	AccomplishMandatory-Grade (0,74) AccomplishMandatory (0,1) CourseViewPct (0,05) AccomplishMandatory-PercentileGrade (0,04) AccomplishMandatory-PctGraded (0,01)

Tabla 4.5: Variables independientes con mayor importancia Gini para clasificar los alumnos en función de las notas de corte seleccionadas.

independientes más influyentes en el rendimiento académico de los alumnos de forma agnóstica al tipo de asignatura. Las variables relacionadas con la evaluación

son las que más influencia tienen, aumentando su importancia según el avanza el instante de predicción. Los pocos accesos a los recursos dentro de la plataforma influyen en la detección de los alumnos en riesgo (umbral 2,5) y suspensos (5,0). Un acceso elevado a la asignatura, en especial en instantes tempranos, detecta alumnos excelentes. Los cuestionarios tienen una influencia limitada a la hora de predecir el rendimiento, y la interacción en los foros de discusión no tiene correlación alguna.

Capítulo 5

Agrupamiento de alumnos

En el capítulo anterior creamos modelos utilizando algoritmos de aprendizaje automático supervisado. La tercera contribución de la presente tesis doctoral tiene como objetivo agrupar a los alumnos sin utilizar una variable clasificadora (*target*) mediante algoritmos no supervisados de agrupamiento (*clustering*), empleando únicamente las variables generadas a partir de las acciones. Estos algoritmos permiten la detección de patrones comunes en los datos de entrada para así formar grupos de alumnos (*clusters*).

Posteriormente, para cada uno de los grupos detectados, analizamos los valores de las variables de sus individuos o instancias para describir las características de cada uno de los grupos. Por último, la cuarta contribución de esta investigación está orientada a analizar si existe alguna relación entre cada uno de los grupos y la evaluación que obtienen los alumnos que lo componen.

5.1. Metodología de generación de modelos de agrupamiento

La metodología utilizada en este proceso de generación de grupos se muestra en la Figura 5.1, indicando con nodos en color naranja aquellas técnicas que se han descartado por no ofrecer mejores resultados. El conjunto de datos de entrada utilizado es el mismo que el descrito en el Capítulo 3, con la excepción de que la variable dependiente que estima el rendimiento del alumno no se ha incluido. En cuanto a los instantes de predicción se mantienen los mismos definidos con anterioridad (10 %, 25 %, 33 % y 50 % de la duración total de la asignatura), por lo que generamos cuatro modelos no supervisados distintos.

Con el objetivo de obtener los mejores resultados y a pesar de que el dataset de partida es el mismo, el tratamiento de las variables es ligeramente distinto que en el caso de los modelos supervisados. En primer lugar, todos los valores vacíos son reemplazados por la media de la variable correspondiente, evitando de esta manera valores nulos [52, 53]. Adicionalmente, con el objetivo de que las variables sean fácilmente comparables entre sí, todas ellas son normalizadas mediante la técnica *z-score*, donde para cada variable su media es cero (μ=0) y su desviación

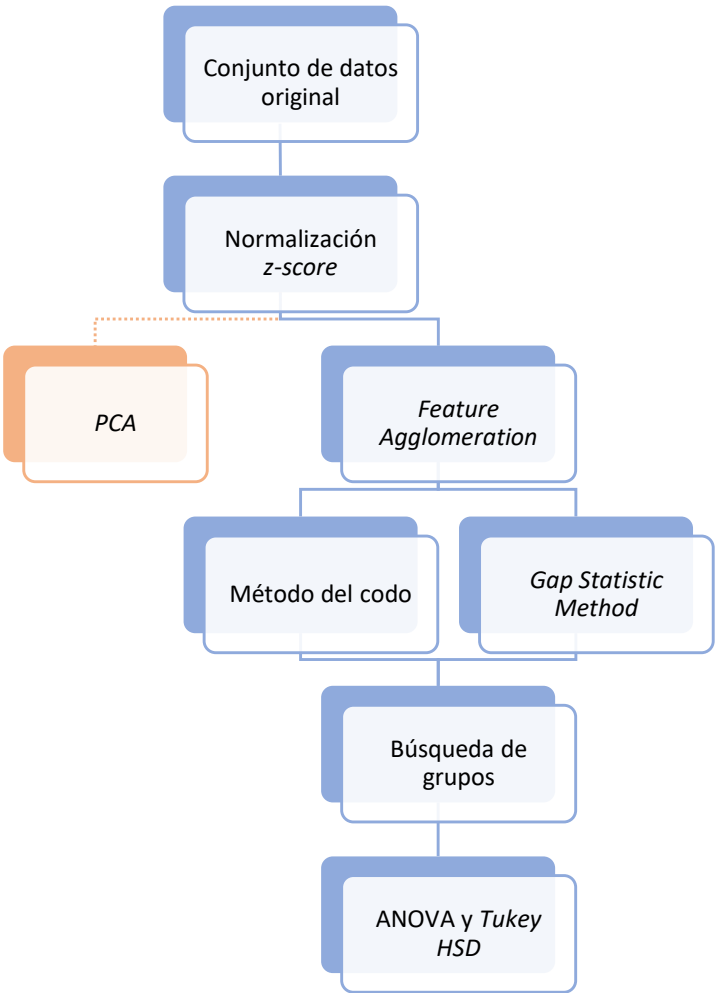

Figura 5.1: Etapas de la metodología seguidas para la construcción de modelos de agrupamiento.

típica es 1 (σ=1).

El conjunto de datos está compuesto de 63 variables, representando un espacio dimensional elevado. Dado que los algoritmos de agrupamiento computan distancias entre todas las instancias, un alto número de dimensiones representa una menor precisión a la hora de generar los grupos [54]. Es por ello por lo que es aconsejable reducir la dimensionalidad del *dataset*, minimizando la pérdida de información del conjunto original [55].

Existen diferentes técnicas para realizar la reducción de dimensionalidad. Al igual que en el caso de los modelos predictivos, hemos probado diferentes técnicas para comprobar con cuál de ellas se obtenían mejores resultados. Los algoritmos de reducción de dimensionalidad probados fueron el análisis de componentes principales (*Principal Component Analysis* o PCA) [56] y *Feature Agglomeration* (FA) [57].

Los resultados obtenidos con el algoritmo PCA mostraban que eran necesarias al menos 20 dimensiones (variables) para explicar el 80 % de la información del dataset original. Sin embargo, con el algoritmo FA se obtiene la misma explicabilidad con únicamente cuatro variables, por lo que seleccionamos este algoritmo

para la reducción de la dimensionalidad. El criterio utilizado para la búsqueda de la agregación de las variables es el WARD [58], métrica que minimiza la varianza de los grupos que está generando. La reducción de las dimensiones usando el algoritmo FA se consigue construyendo nuevas variables a partir de combinaciones de las variables originales.

El resultado de la aplicación del algoritmo FA proporciona cuatro nuevas dimensiones. En función de la naturaleza de las variables que agrega cada una de estas nuevas dimensiones se describen como (para más detalles consultar el Anexo D):

- *URL and Assignment Access* (`UAA`): esta variable agrega los instantes de tiempo en los que se consultan recursos externos y en los que se visualizan y entregan las tareas evaluables.
- *Mandatory and Optional Assignment evaluation* (`MOA`): agregación de todas las variables relacionadas con las evaluaciones de los alumnos, tanto obligatorias como opcionales.
- *Quiz Access Time* (`QAT`): agregación de las variables relacionadas con los instantes de acceso, comienzo y finalización de los cuestionarios.
- *Course and Resource View* (`CIR`): esta dimensión agrega las variables relacionadas con el instante de acceso a las asignaturas, así como a los recursos internos proporcionados en la asignatura.

Una vez realizada la reducción de dimensionalidad del *dataset*, el siguiente paso es aplicar un algoritmo de agrupamiento para obtener grupos de alumnos. Hemos seleccionado el algoritmo *k-means* por ser uno de los más comunes para resolver este tipo de problemas. Una de las limitaciones de este algoritmo es que necesita recibir como parámetro el número de grupos a generar, por lo que es necesario encontrar el número de grupos óptimo (compromiso entre un número de grupos y la distancia media entre las instancias y el centro del grupo al que pertenecen).

Para determinar el valor óptimo del número de grupos se utilizaron dos técnicas, obteniéndose en ambos el mismo resultado. En primer lugar, se utilizó el método del codo (*elbow method*) que se basa en utilizar una visualización para encontrar un equilibrio entre el número de grupos y la distancia media entre los registros dentro de un mismo grupo [59, 60]. La visualización del método del codo no siempre ofrece un resultado claro y es necesario tomar decisiones adicionales para resolver ambigüedades [61]. Para evitar esa toma de decisiones existen otros algoritmos como *gap statistic method* que permite calcular el número óptimo de grupos sin recurrir a una visualización [62, 63]. Tras aplicar ambos enfoques, el número óptimo de grupos encontrados fue de seis en todos los instantes de tiempo analizados.

Por último, los hiperparámetros del algoritmo *k-means* utilizados, además del número de grupos óptimo, fueron la inicialización de centroides de forma aleatoria y el uso de distancia Euclídea.

Tras la obtención de los grupos realizamos un análisis ANOVA para conocer sí

existen diferencias significativas entre los valores de las variables agregadas para cada uno de los grupos. Tras el ANOVA también se ejecutaron *tests post-hoc Tukey* (*Tukey's Honest Significant Difference* o *Tukey's HSD*) para identificar exactamente qué variables eran diferentes entre sí en los distintos grupos. Esto nos permitió identificar las variables que definen a cada grupo, pudiendo definir las peculiaridades de cada uno de los grupos.

Todo el proceso descrito se ha realizado para cada uno de los instantes de análisis definidos en base a la duración de la asignatura (10 %, 25 %, 33 % y 50 %), manteniendo así el criterio de identificación temprana de los grupos.

5.2. Resultados de los modelos de agrupamiento

Tanto el método del codo como el *gap statistic method* mostraron que el número óptimo de grupos es seis, independientemente del instante de análisis. Esta búsqueda se realizó buscando el número óptimo de grupos entre 2 y 10. La Figura 5.2 y la Tabla 5.1 detallan la distribución de cada una de las cuatro variables agregadas en los diferentes grupos obtenidos por *k-means*, para los distintos instantes de análisis.

		Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6
10 %	N	12.162	1.691	837	756	4.198	5.616
	UAA	0,0284 ± 0,256	-1,1638 ± 0,608	0,2016 ± 0,543	-0,1940 ± 0,700	0,2811 ± 0,530	0,0748 ± 0,386
	MOA	-0,1506 ± 0,328	0,3153 ± 0,471	0,5061 ± 0,729	0,6515 ± 0,708	-0,0237 ± 0,345	0,0859 ± 0,412
	QAT	-0,0021 ± 0,174	-0,0092 ± 0,389	2,3969 ± 0,873	-2,6879 ± 1,194	0,0328 ± 0,250	-0,0127 ± 0,286
	CIR	0,0841 ± 0,250	-0,6530 ± 0,465	0,1862 ± 0,807	-0,4885 ± 0,590	1,1991 ± 0,489	-0,8440 ± 0,331
25 %	N	7.088	1.897	8.600	3.353	1.315	3.007
	UAA	0,1469 ± 0,379	-0,0436 ± 0,545	0,0658 ± 0,329	-0,7834 ± 0,416	0,1455 ± 0,504	0,3028 ± 0,503
	MOA	-0,0010 ± 0,389	0,4188 ± 0,641	-0,1851 ± 0,381	0,2586 ± 0,435	0,2128 ± 0,570	-0,1140 ± 0,359
	QAT	0,0153 ± 0,274	-1,6652 ± 0,580	0,0066 ± 0,212	0,0002 ± 0,274	2,1413 ± 0,799	0,0591 ± 0,324
	CIR	-0,5945 ± 0,269	-0,3984 ± 0,474	0,2397 ± 0,298	-0,5274 ± 0,356	0,1075 ± 0,799	1,5082 ± 0,556
33 %	N	4.134	7.438	2.460	1.452	7.266	2.510
	UAA	-0,6901 ± 0,392	0,0702 ± 0,331	-0,0471 ± 0,507	0,1817 ± 0,524	0,1800 ± 0,387	0,3482 ± 0,527
	MOA	0,2394 ± 0,429	-0,2061 ± 0,406	0,3432 ± 0,607	0,1563 ± 0,519	-0,0245 ± 0,394	-0,1393 ± 0,390
	QAT	0,0166 ± 0,262	0,0095 ± 0,215	-1,4321 ± 0,488	2,0941 ± 0,754	0,0180 ± 0,266	0,0846 ± 0,357
	CIR	-0,4895 ± 0,329	0,3148 ± 0,320	-0,3755 ± 0,433	0,1150 ± 0,799	-0,5191 ± 0,263	1,6775 ± 0,626
50 %	N	3.055	8,051	1.520	1.627	4.957	6.050
	UAA	-0,0712 ± 0,454	0,1980 ± 0,382	0,4268 ± 0,586	0,1841 ± 0,518	0,1479 ± 0,421	-0,5054 ± 0,409
	MOA	0,2717 ± 0,488	-0,2022 ± 0,414	-0,2664 ± 0,433	0,0859 ± 0,537	-0,0985 ± 0,401	0,2563 ± 0,452
	QAT	-1,2791 ± 0,422	0,0112 ± 0,242	0,1999 ± 0,559	1,9876 ± 0,758	0,0109 ± 0,265	0,0374 ± 0,266
	CIR	-0,3566 ± 0,371	-0,3543 ± 0,293	2,1397 ± 0,671	0,0930 ± 0,734	0,6423 ± 0,346	-0,4374 ± 0,301

Tabla 5.1: Número de estudiantes por grupo (N), y valor de la variable agregada ± desviación típica para cada uno de los grupos, para los cuatro instantes de análisis (10 %, 25 %, 33 % y 50 %).

Para las distintas variables, la aplicación del análisis ANOVA encontró diferencias significativas entre los grupos (valor $p < 0{,}05$) en cada uno de los instantes de análisis. Con este resultado es posible aplicar los *tests Tukey post-hoc* para analizar las diferencias entre grupos de los valores de cada variable agregada [64]. La Figura 5.3 muestra los resultados de esos *tests*, donde los bigotes representan el intervalo de confianza 95 %, las líneas puntadas indican el valor medio de la variable y Gn representa el número de grupo. Estos resultados pueden ser resumidos del siguiente modo:

Figura 5.2: Media de las variables por grupo en los diferentes instantes de tiempo. Los bigotes representan la desviación típica.

- Instante 10 % de la asignatura: las variables `UAA`, `MOA` y `CIR` muestran diferencias significativas en todos los grupos (valor $p < 0{,}05$ para los *tests de Tukey*). Sin embargo, para la variable `QAT` no existe una diferencia significativa entre los grupos 1, 2 y 6.

- Instante 25 % de la asignatura: las variables `MOA` y `CIR` muestran diferencias significativas en todos los grupos. Los resultados para la variable `UAA` indican que no existen diferencias significativas entre los grupos 1 y 5, mientras que en el caso de la variable `QAT` no se aprecian diferencias entre los grupos 1, 3 y 4.

- Instante 33 % de la asignatura: todos los grupos muestran diferencias significativas para las variables `MOA` y `CIR`. En cuanto a la variable `UAA`, no existen diferencias entre los grupos 4 y el 5. La variable `QAT` muestra los mismos valores, no existiendo diferencias significativas en los grupos 1, 2 y 5.

- Instante 50 % de la asignatura: en este instante de predicción ninguna variable muestra diferencias significativas para todos los grupos. Los grupos 2

y 4 muestran valores similares para la variable `UAA`; los grupos 1 y 6 para la variable `MOA`; los grupos 2 y 5 para la variable `QAT` y por último los grupos 1 y 2 para la variable `CIR`.

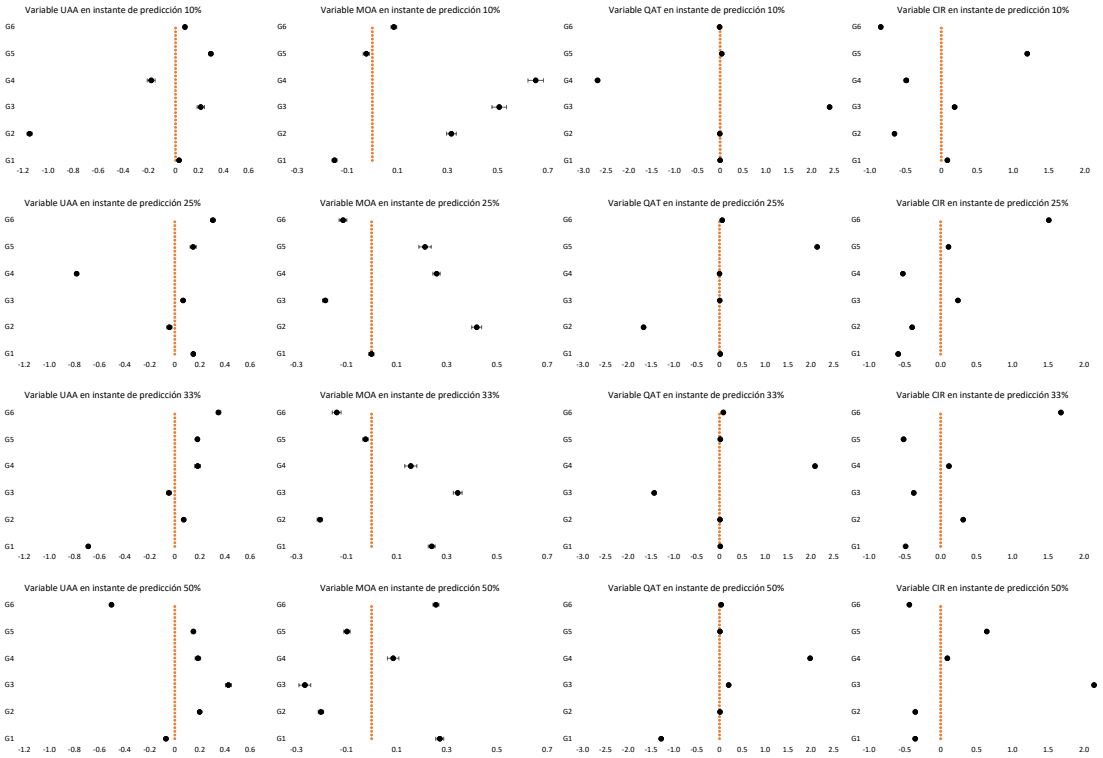

Figura 5.3: Valores de los resultados de los *test Tukey post-hoc* para las variables y grupos para los diferentes instantes de análisis.

5.3. Discusión de los resultados de los modelos de agrupamiento

En primer lugar, cabe recordar que los conjuntos de datos utilizados para cada uno de los instantes de tiempo son diferentes, provocando que cada uno de los conjuntos de grupos sean diferentes en cada instante. Por tanto, es necesario analizar las características de cada uno de los grupos obtenidos en cada instante y comprobar si existen similitudes a través de los distintos instantes de análisis.

Con este propósito se analizó cómo los valores de cada variable determinan la pertenencia a cada uno de los grupos (*test de Tukey*). Adicionalmente, también se estudió si esta pertenencia a los grupos se repetía en cada uno de los instantes de análisis. En caso afirmativo significaría que el algoritmo de agrupamiento genera grupos similares en los distintos instantes de tiempo con diferentes conjuntos de datos de entrada.

Dado que las variables están normalizadas con $\mu = 0$ y $\sigma = 1$ (Sección 5.1), se han tenido en cuenta los siguientes umbrales para analizar las variables:

- Calificamos como valor muy bajo a aquél que está por debajo de -1.
- Un valor bajo es aquél que está entre -1 y -0,5 (no incluido).

- Un valor medio es aquél que está entre -0,5 y 0,5 (ambos incluidos).
- Valor alto es el que está entre 0,5 (no incluido) y 1.
- Un valor muy alto supera el valor de 1.

Analizando los valores de las variables con estos criterios para los grupos encontrados, se pueden identificar grupos diferentes que se repiten a lo largo de los instantes de tiempo.

- Variable `QAT`. Como muestra la Figura 5.3, existen dos grupos con un valor extremo de esta variable. Este patrón se repite en cada uno de los instantes de análisis estudiados.
  - Grupo `QAT`↓↓. Un valor muy bajo de esta variable define el grupo de alumnos que responde a los cuestionarios significativamente antes que el resto de los compañeros. La Tabla 5.2 muestra que los grupos 4, 2, 3 y 1 tienen en común esta condición para los instantes 10 %, 25 %, 33 % y 50 % respectivamente.
  - Grupo `QAT`↑↑. Este otro grupo muestra un valor muy alto de esta variable, mostrando estudiantes que responden a los cuestionarios significativamente más tarde que el resto de los compañeros.
- Grupo `UAA`↓. Valores bajos y muy bajos de esta variable define otro de los grupos identificados en todos los instantes de tiempo (Figura 5.3). Este grupo representa a los alumnos que acceden a los recursos externos y a las tareas dentro del LMS significativamente más temprano que sus compañeros.
- Grupo `CIR`↑↑. La Figura 5.3 muestra, para los distintos instantes de tiempo, un grupo con valores muy elevados de la variable CIR. Los estudiantes incluidos dentro de este grupo procrastinan, tardando significativamente más que el resto de los alumnos en acceder a la asignatura y a los recursos proporcionados por los profesores dentro de la plataforma.
- Grupo `MOA`↓`CIR`↑. Existe un grupo en todos los instantes de tiempo donde la variable de evaluación está por debajo de la media (MOA) y el acceso a los recursos de la asignatura por encima de la media (CIR). Este grupo contiene estudiantes definidos por esas variables pero que no cumplen las condiciones para pertenecer a los grupos mencionados con anterioridad.
- *Grupo Promedio*. Por último, existe un grupo en los distintos instantes que contiene aquellos alumnos que tienen valores medios en todas las variables. Es por ello por lo que no existe una variable o combinación de variables que permita definir el grupo.

Tras esta discusión, podemos concluir que hemos alcanzado la **Contribución 3**: los análisis realizados sobre los resultados del algoritmo de agrupamiento muestran que existen seis grupos diferentes que se repiten en todos los instantes analizados. Estos grupos agrupan a los alumnos en base a las acciones que han realizado hasta el instante en el que se realiza el análisis, sin tener en cuenta características propias de las asignaturas. Es importante remarcar que, curiosamente, la variable que agrupa las evaluaciones de los alumnos hasta el instante de análisis (MOA) no definen ningún grupo por sí sola. La Tabla 5.2 muestra la

Instante de análisis	QAT↓↓	QAT↑↑	UAA↓	CIR↑↑	MOA↓CIR↑	*Grupo Promedio*
10 %	G4	G3	G2	G5	G1	G6
25 %	G2	G5	G4	G6	G3	G1
33 %	G3	G4	G1	G6	G2	G5
50 %	G1	G4	G6	G3	G5	G2

Tabla 5.2: Correspondencia entre los grupos para un instante de análisis particular y los grupos detectados para cualquier instante (primera fila). G_n representa el grupo n.

correspondencia entre los grupos identificados en los distintos instantes de análisis y los seis grupos de alumnos repetidos en todos los instantes.

5.4. Correlación entre el agrupamiento y el rendimiento académico

La sección anterior detectó diferentes patrones de comportamiento de interacción con el LMS, constituyendo seis grupos que se repiten en cualquier instante del análisis. Esta sección parte de los resultados obtenidos en la anterior para analizar si existe alguna dependencia entre los grupos y el rendimiento de los estudiantes. Para ello, utilizamos la variable continua que estima el rendimiento del alumno tal y como se explica en la Sección 3.6.

De cara a analizar esta dependencia se utilizó el análisis ANOVA, comprobando que sí existía esa relación entre la pertenencia al grupo y la evaluación (valor $p <$ 0,05). Posteriormente se ejecutaron *tests Tukey's HSD post-hoc* con el objetivo de observar, para cada par de grupos, si existían diferencias significativas en el rendimiento académico de los alumnos pertenecientes ambos grupos.

La Figura 5.4 muestra los resultados de los *tests Tukey's HSD*, siendo los bigotes el intervalo de confianza 95 % y las líneas punteadas el valor medio de la variable. En esta figura, cuando dos intervalos de confianza (bigotes) se solapan, significa que no existe diferencia significativa entre los grupos que representan [65]. Se detectaron las siguientes relaciones entre los grupos y el rendimiento académico de sus alumnos:

- El rendimiento académico más elevado es obtenido por aquellos alumnos que acceden a los cuestionarios y los realizan significativamente antes que el resto de los compañeros (QAT↓↓). En algunos instantes de análisis, esta diferencia no es significativa con otros grupos.
- Los estudiantes con menor rendimiento académico pertenecen al grupo que muestra procrastinación a la hora de acceder a los recursos proporcionados por los profesores y con una evaluación en las tareas por debajo de la media (MOA↓CIR↑). Este grupo muestra diferencias significativas con el resto de los grupos, excepto con el siguiente para el instante 50 % de la asignatura.
- Los estudiantes con una procrastinación muy elevada en el acceso al curso

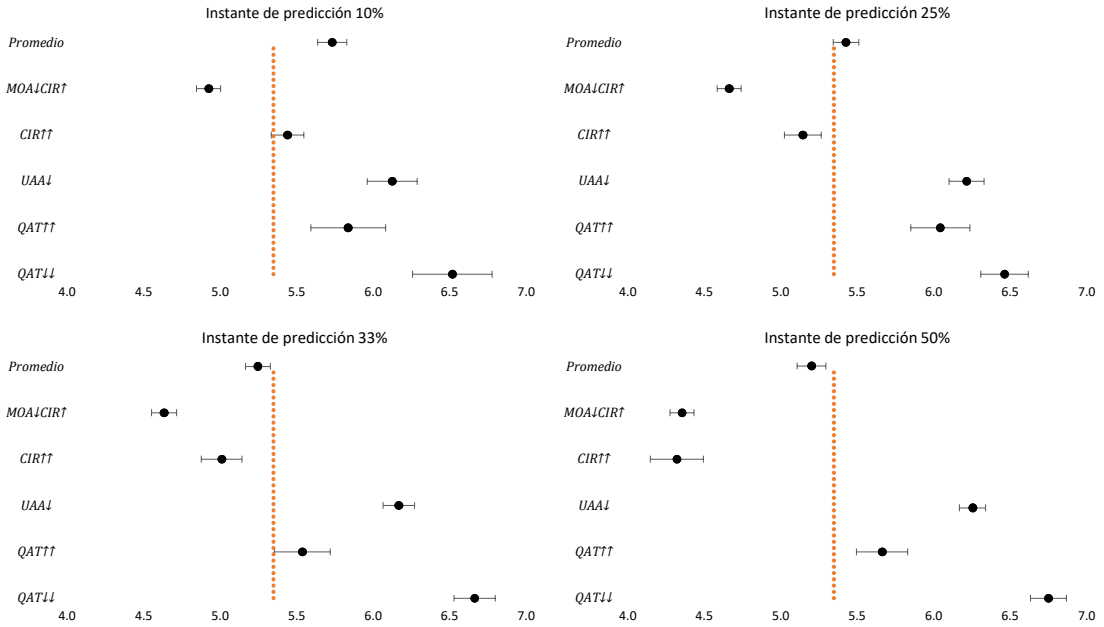

Figura 5.4: Rendimiento de los alumnos para cada uno de los grupos identificados en los distintos instantes de análisis.

y a los recursos propuestos por los profesores (CIR↑↑) obtienen el segundo peor rendimiento. En el caso del instante 50 % no existe diferencia con el grupo anterior.

- El grupo UAA↓ contiene a los alumnos que obtienen el segundo mejor rendimiento académico. Estos alumnos se corresponden con aquéllos que visualizan los recursos externos y las tareas significativamente antes que el resto de los compañeros.
- Por último, los grupos restantes, QAT↑↑ y el *Grupo Promedio*, contienen a alumnos que obtienen rendimientos próximos a la media en todos los instantes. En algunos de los instantes de análisis no existe diferencia significativa entre ellos.

Por tanto, nuestra **Contribución 4** concluye que cuatro de los seis grupos identificados para cualquier instante de análisis están directamente relacionados con la evaluación de los alumnos que contienen. Esta correlación es válida para cualquier asignatura, independientemente de la metodología de estudio, área, disciplina y duración.

Capítulo 6

Relación entre la procrastinación y el rendimiento académico

Las cuatro primeras contribuciones de esta tesis doctoral están orientadas a analizar la información obtenida de las acciones de los estudiantes en un LMS para un número elevado de asignaturas de distinta naturaleza. El beneficio de ese estudio radica en la independencia del tipo de asignatura, ofreciendo un elevado grado de generalización. De este modo, las conclusiones obtenidas son aplicables a cualquier tipo de asignatura que utilice una plataforma de aprendizaje para evaluar tareas opcionales y obligatorias del alumnado.

La utilización de un elevado número de datos pertenecientes a múltiples asignaturas también posee inconvenientes. El primero es no disponer de la nota final de los alumnos en las asignaturas debido a limitaciones técnicas y de protección de datos. El segundo es que la exactitud de modelos de aprendizaje automático será potencialmente menor que el estudio de un tipo de asignatura en particular [20].

Este capítulo está enmarcado dentro de un escenario distinto. Nos centramos en el análisis de los registros de un LMS para una asignatura en particular, tratando de detectar comportamientos de procrastinación y su asociación con la evaluación real obtenida por los estudiantes [66]. Esta detección y asociación se realiza en instantes intermedios de la asignatura.

Las siguientes secciones describen desde la asignatura seleccionada para el estudio hasta los resultados obtenidos, pasando por las variables predictoras definidas y los algoritmos seleccionados. La Figura 6.1 muestra los pasos seguidos en este estudio.

6.1. Selección de la asignatura

Para esta contribución, al igual que en las anteriores, se utilizan datos sobre las acciones de los alumnos dentro de la plataforma Moodle. El estudio utiliza información de 33 alumnos, 20 hombres y 13 mujeres, matriculados en una asignatura perteneciente al cuarto curso del Grado de Geomántica y Topología de la Universidad de Oviedo. Esta asignatura tiene una duración semestral, está im-

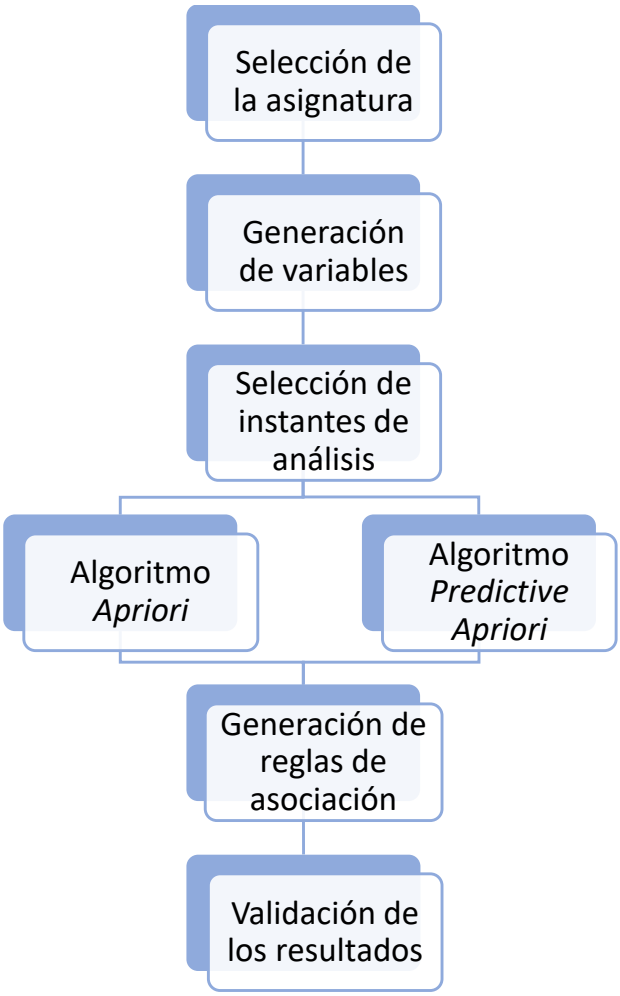

Figura 6.1: Etapas de la metodología seguidas para el análisis de la procrastinación y su relación con el rendimiento académico de los alumnos.

partida de forma presencial y utiliza la plataforma de aprendizaje para la entrega de tareas y la distribución de contenidos.

La asignatura está dividida en tres unidades didácticas que comparten la misma estructura. Cada una de ellas contiene los recursos internos y externos relacionados con una temática específica, tareas a realizar y entregar por parte de los alumnos y cuestionarios. En el caso de que los alumnos quieran optar a una evaluación continua, deben entregar una o más tareas en cada una de las unidades didácticas y completar los cuestionarios.

La elección de esta asignatura radica en la disponibilidad de la nota final de los alumnos y su elevada interacción con la plataforma. Los registros de las acciones están anonimizados para cumplir con las exigencias legales. No obstante, esta información puede ser unida con las notas reales a través de un identificador que, si bien no permite reconocer a los alumnos, nos sirve para vincular la información proveniente de varias fuentes de datos.

6.2. Generación de variables

En este estudio analizamos las variables de relacionadas con la procrastinación y su asociación con la nota real de los estudiantes. Los datos en crudo correspondientes a las acciones de los estudiantes, al igual que en los estudios anteriores de predicción y de agrupamiento, están almacenados dentro de la base de datos de Moodle en diversas tablas (`mdl_log`).

A partir de las acciones, se han construido las variables mostradas en la Tabla 6.1, siendo todas ellas numéricas. Estas variables pueden clasificarse dentro de los siguientes grupos diferentes en base a su naturaleza:

- Variables calculadas como contadores de acceso, donde se incluyen las variables `Resources`, `Urls` y `Assignments` que miden el número de accesos a recursos internos, recursos externos y tareas respectivamente.
- Variables de completitud de una tarea, que únicamente incluye la variable `Quizzes`. Mide el tiempo empleado en completar un cuestionario.
- Variables correspondientes a la procrastinación. En la tabla se visualizan con el prefijo `Timeto` y miden el tiempo hasta realizar cada una de las acciones que definen.
- Variable `Grade` que muestra la evaluación real (rendimiento) de los alumnos en la asignatura.

Variable	Descripción
`Resources`	Número de accesos a recursos internos de la plataforma.
`Urls`	Número de accesos a recursos externos a la plataforma.
`Assignments`	Número de accesos a tareas entregables.
`Quizzes`	Tiempo empleado en completar los cuestionarios (segundos).
`Timetoresources`	Tiempo hasta el primer acceso a un recurso interno.
`Timetourls`	Tiempo hasta el primer acceso a un recurso externo.
`Timetoassigns`	Tiempo hasta el primer acceso a una tarea.
`Timetoquizzes`	Tiempo hasta el primer acceso a un cuestionario.
`Timetofirst`	Tiempo hasta el primer acceso a la unidad.
`Grade`	Evaluación real del alumno en escala 0-10.

Tabla 6.1: Variables usadas en el estudio con su descripción.

Debido a que los algoritmos utilizados (véase Sección 6.4) requieren de variables categóricas en lugar de numéricas, realizamos un proceso de discretización. Las variables numéricas fueron convertidas en discretas utilizando grupos de igual número de instancias (*equal-frequency*) [67] para tres grupos (`bajo`, `medio`, `alto`) [68]. La única excepción a esta regla es la variable `Nota` para la que se ha realizado una discretización manual [67] considerando una evaluación `baja` aquella que está por debajo de la media. Por último, la división entre la evaluación `media` y `alta` se realizó usando el método *equal-width* entre los alumnos con una evaluación por encima de la media.

6.3. Instantes de análisis

La estructura repetitiva de la asignatura en unidades didácticas similares permite la realización de análisis en instantes previos a su finalización. Los instantes en los que se realiza el análisis se corresponden con la finalización de cada una de las tres unidades didácticas. Adicionalmente también se realiza un análisis al finalizar la asignatura con toda la información de los estudiantes ya disponible.

Para cada uno de los instantes se tiene únicamente en cuenta la información de la unidad didáctica que se acaba de finalizar, mientras que para el análisis al finalizar el curso se tienen en cuenta la información obtenida a través de las tres unidades didácticas, toda la información de la asignatura.

6.4. Algoritmos utilizados

Estudios previos [69] muestran cómo las reglas de asociación pueden ser útiles a la hora de buscar relaciones entre el comportamiento de los alumnos y su rendimiento académico. Esto, sumado al bajo número de alumnos, hace que el uso de algoritmos predictivos esté desaconsejado, al no tener un número suficiente de instancias para hacer una predicción con un buen rendimiento.

Las reglas de asociación se utilizan en problemas de minería de datos con el objetivo de obtener asociaciones, patrones frecuentes y correlaciones entres los valores de los atributos que describen el conjunto de datos [70]. Dado un *dataset* formado por conjunto de atributos $A = \{a_1, a_2, ...a_n\}$, una regla de asociación se define como:

$$X \Rightarrow Y \quad donde \;\; X, Y \subseteq A \quad y \quad X \cap Y = \emptyset \tag{6.1}$$

De esta forma, una regla de asociación podría tener una estructura como 'SI $atributo_1$ Y $atributo_2$ ENTONCES $atributo_3$ Y $atributo_4$'. En la regla anterior $atributo_1$ y $atributo_2$ son los predecesores o antecedentes, mientras que $atributo_3$ y $atributo_4$ con los sucesores o consecuentes.

En este estudio se ha utilizado Weka [71] como software para la obtención de las reglas de asociación. Para ello, se han empleado las implementaciones de los algoritmos *Apriori* [72] y *Predictive Apriori* [73]. La selección de dos algoritmos viene motivada por estudios previos que muestran cómo este segundo algoritmo puede mejorar los resultados obtenidos por *Apriori* [74].

Dada una regla de asociación, la cobertura o soporte (*support*), la confianza o precisión (*confidence*) y *predictive accuracy* son medidas para medir la calidad de las reglas. La cobertura (Ecuación 6.2) mide la proporción de instancias que la regla cubre, mientras que la confianza (Ecuación 6.3) indica la proporción de veces que la regla se cumple cuando ésta se puede aplicar (cuántas veces el consecuente es cierto, tras serlo el antecedente). Por último, la métrica *predictive accuracy* proporciona un valor corregido de la confianza en función de la cobertura en base a probabilidades condicionales [73].

$$cobertura(X \Rightarrow Y) = \frac{n^{o}\ de\ instancias\ que\ contienen\ X\ e\ Y}{n^{o}\ total\ de\ instancias} \tag{6.2}$$

$$confianza(X \Rightarrow Y) = \frac{n^{o}\ de\ instancias\ que\ contienen\ X\ e\ Y}{n^{o}\ de\ instancias\ que\ contienen\ X} \tag{6.3}$$

El algoritmo *Apriori* extrae conjuntos de elementos frecuentes del conjunto de datos para posteriormente generar reglas de asociación. Este algoritmo emplea un enfoque iterativo de búsqueda por nivel donde se utilizan k conjuntos de elementos para explorar $(k+1)$ conjuntos de elementos. En cada iteración hay dos fases. Primero se genera un conjunto de datos con los elementos candidatos. El segundo paso cuenta la ocurrencia de cada conjunto de candidatos (soporte) y se realiza la poda de aquellos conjuntos con menor frecuencia. Al finalizar cada iteración, se incrementa en uno el tamaño de los grupos de candidatos. La búsqueda finaliza cuando no se encuentran nuevos candidatos.

El algoritmo *Predictive Apriori* busca, con un umbral de soporte creciente, las mejores n reglas relativas al valor del *predictive accuracy*. En este caso, una regla se añade sí y solo sí la métrica *predictive accuracy* esperada se encuentra entre las n mejores y no está contenida por una regla con al menos el mismo valor esperado de dicha métrica.

6.5. Resultados de las reglas de asociación

Dado que el objetivo es analizar el impacto de la procrastinación en el rendimiento de los alumnos, únicamente se han seleccionado aquellas reglas que relacionan cualquier conjunto de variables con la evaluación del alumno.

En el caso del algoritmo *Apriori*, seleccionamos aquellas reglas que tienen una confianza superior a 0,95 con una cobertura mínima del 0,1. En el caso del algoritmo *Predictive Apriori* hemos tenido en cuenta aquellas reglas con un *predictive accuracy* superior a 0,95. La Tabla 6.2 muestra el número de reglas que se han obtenido con cada algoritmo en cada instante de análisis (el Anexo E detalla todas de las reglas obtenidas).

Unidad Didáctica	***Apriori***	***Predictive Apriori***
Unidad Didáctica 1	12	47
Unidad Didáctica 2	6	52
Unidad Didáctica 3	19	53
Final asignatura	7	51

Tabla 6.2: Número de reglas obtenido para cada uno de los instantes de análisis y cada uno de los algoritmos utilizados.

La Tabla 6.3 muestra para cada algoritmo y en los distintos instantes de análisis seleccionados las seis reglas con mayor cobertura. De esta forma es posible comparar los resultados de ambos algoritmos en cada una de las unidad didácticas de una forma sencilla.

		Antecedente	Consecuente	Cobertura
Unidad Didáctica 1	*Apriori*	Quizzes = medio ∧ Timetoquizzes = medio	Grade = alta	0,182
		Resources = alto ∧ Timetourls = bajo	Grade = alta	0,152
		Resources = medio ∧ Quizzes = medio	Grade = alta	0,121
		Resources = alto ∧ Timetofirsts = medio	Grade = alta	0,121
		Quizzes = medio ∧ Timetoresources = medio	Grade = alta	0,121
		Resources = alto ∧ Urls = alto ∧ Timetourls = bajo	Grade = alta	0,121
	Predictive Apriori	Quizzes = medio ∧ Timetoquizzes = medio	Grade = alta	0,182
		Resources = alto ∧ Timetourls = bajo	Grade = alta	0,152
		Resources = medio ∧ quizzes = medio	Grade = alta	0,121
		Resources = alto ∧ Timetofirsts = medio	Grade = alta	0,121
		Quizzes = medio ∧ Timetoresources = medio	Grade = alta	0,121
		Timetoquizzes = bajo ∧ Timetoresources = bajo ∧ Timetoassigns = bajo	Grade = alta	0,121
Unidad Didáctica 2	*Apriori*	Timetoresources = medio ∧ Timetourls = medio	Grade = baja	0,152
		Resources = medio ∧ Timetourls = bajo	Grade = media	0,121
		Resources = alto ∧ Timetoquizzes = bajo	Grade = alta	0,121
		Resources = alto ∧ Timetoresources = bajo	Grade = alta	0,121
		Timetoresources = medio ∧ Timetourls = bajo	Grade = media	0,121
		Timetoresources = medio ∧ Timetourls = medio ∧ Timetoassigns = alto	Grade = baja	0,121
	Predictive Apriori	Timetoresources = medio ∧ Timetourls = medio	Grade = baja	0,152
		Resources = medio ∧ Timetourls = bajo	Grade = media	0,121
		Resources = alto ∧ Timetoquizzes = bajo	Grade = alta	0,121
		Resources = alto ∧ Timetoresources = bajo	Grade = alta	0,121
		Timetoresources = medio ∧ Timetourls = bajo	Grade = media	0,121
		Resources = alto ∧ Quizzes = bajo	Grade = baja	0,091
Unidad Didáctica 3	*Apriori*	Quizzes = bajo ∧ Timetoassigns = alto	Grade = baja	0,182
		Resources = alto ∧ Quizzes = bajo	Grade = baja	0,152
		Quizzes = bajo ∧ Timetoresources = alto	Grade = baja	0,152
		Quizzes = bajo ∧ Timetofirsts = alto	Grade = baja	0,152
		Timetoquizzes = alto ∧ Timetofirsts = alto	Grade = baja	0,152
		Resources = alto ∧ Quizzes = bajo ∧ Timetoquizzes = alto	Grade = baja	0,152
	Predictive Apriori	Quizzes = bajo ∧ Timetoassigns = alto	Grade = baja	0,182
		Resources = alto ∧ Quizzes = bajo	Grade = baja	0,152
		Quizzes = bajo ∧ Timetoresources = alto	Grade = baja	0,152
		Quizzes = bajo ∧ Timetofirsts = alto	Grade = baja	0,152
		Timetoquizzes = alto ∧ Timetofirsts = alto	Grade = baja	0,152
		Resources = bajo ∧ Quizzes = bajo	Grade = baja	0,121

Tabla 6.3: Reglas de asociación con mayor cobertura obtenidas para ambos algoritmos tras la finalización de cada una de las unidades didácticas.

Del mismo modo, se analizan las reglas obtenidas al finalizar la asignatura. Como se ha mencionado anteriormente, este análisis tiene en cuenta todas las acciones que los alumnos realizan en la duración total de la asignatura. La Tabla 6.4 muestra las reglas con mayor cobertura al igual que en el análisis anterior.

Por último también analizamos la existencia de reglas que se repiten en distintos instantes de análisis. La Tabla 6.5 muestra las reglas de asociación que se repiten en al menos dos instantes de análisis previos a la finalización de la

	Antecedente	Consecuente	Cobertura
Apriori	`Quizzes` = `medio` ∧ `Timetofirst` = `alto`	`Grade` = `baja`	0,152
	`Urls` = `bajo` ∧ `Quizzes` = `bajo`	`Grade` = `baja`	0,121
	`Urls` = `bajo` ∧ `Timetoassigns` = `alto`	`Grade` = `baja`	0,121
	`Timetoresources` = `alto` ∧ `Timetoassigns` = `alto`	`Grade` = `baja`	0,121
	`Quizzes` = `alto` ∧ `Timetoresources` = `bajo` ∧ `Timetoassigns` = `bajo`	`Grade` = `alta`	0,121
	`Quizzes` = `alto` ∧ `Timetoassigns` = `alto` ∧ `Timetofirst` = `bajo`	`Grade` = `alta`	0,121
Predictive Apriori	`Quizzes` = `medio` `Timetofirst` = `alto`	`Grade` = `baja`	0,152
	`Urls` = `bajo` ∧ `Quizzes` = `bajo`	`Grade` = `baja`	0,121
	`Urls` = `bajo` ∧ `Timetoassigns` = `alto`	`Grade` = `baja`	0,121
	`Timetoresources` = `alto` ∧ `Timetoassigns` = `alto`	`Grade` = `baja`	0,121
	`Quizzes` = `alto` ∧ `Timetoresources` = `bajo` ∧ `Timetoassigns` = `bajo`	`Grade` = `alta`	0,121
	`Quizzes` = `alto` ∧ `Timetoassigns` = `alto` ∧ `Timetofirst` = `bajo`	`Grade` = `alta`	0,121

Tabla 6.4: Reglas de asociación con mayor cobertura obtenidas para ambos algoritmos tras la finalización de la asignatura.

asignatura, independientemente del algoritmo. Para cada aparición se incluye la cobertura asociada a esa regla en ese algoritmo y ese instante de análisis.

6.6. Discusión de los resultados de las reglas de asociación

La Tabla 6.3 y la Tabla 6.4 muestran las 6 reglas con mayor cobertura obtenidas, agrupadas por algoritmo. El 83 % de las reglas (30 de 36) obtenidas en los análisis de las unidades didácticas (Tabla 6.3) tienen en cuenta al menos una variable relacionada con la procrastinación. Este mismo análisis para las reglas obtenidas tras finalización de la asignatura (Tabla 6.4) muestra el mismo resultado, con un 83 % de reglas que incluye al menos una variable de procrastinación (10 de 12).

Analizando las reglas que contienen variables de procrastinación, es posible comprobar que cuando un alumno accede de forma temprana a recursos, cuestionarios, etc. (variables con prefijo `Timeto` con valor igual a `bajo`) obtiene una calificación `alta`. Sin embargo, existen cuatro reglas (13 %) donde el acceso a los recursos externos (`Timetourls`) de forma temprana no implica la obtención de una evaluación `alta` sino `media`.

Por contra, cuando las variables de procrastinación tienen un valor elevado (variables con prefijo `Timeto` con valor igual a `alto`), la evaluación de los alumnos es siempre `baja`. Con estos resultados se puede concluir que aquellos alumnos que procrastinan a la hora de realizar acciones dentro de la plataforma de aprendizaje tienen una mayor probabilidad de obtener un rendimiento bajo en la asignatura.

Las reglas mostradas en la Tabla 6.3 permiten también comprobar que los re-

sultados obtenidos por ambos algoritmos son similares. Concretamente, las cinco primeras reglas obtenidas con cada algoritmo son idénticas en todos los instantes de análisis, usando como métrica de ordenación la cobertura para el algoritmo *Apriori* y el *predictive accuracy* para el algoritmo *Predictive Apriori*. En el caso de las reglas obtenidas tras la finalización de la asignatura, ambos algoritmos obtienen exactamente las mismas reglas, manteniendo el mismo criterio de ordenación.

Por otro lado, la Tabla 6.5 muestra las reglas de asociación que se han obtenido en al menos dos instantes de análisis previos a la finalización de la asignatura, independientemente del algoritmo. Trece reglas están repetidas en al menos dos instantes de tiempo, de las cuales únicamente cuatro de ellas aparecen en la Tabla 6.3. Esto indica que el resto reglas repetidas no obtienen una cobertura elevada. Por tanto, existen reglas comunes en los distintos instantes de análisis, pero solo el 30,8 % (4 de 13) de ellas están entre las más influyentes para identificar el rendimiento de los alumnos.

Dentro de la Tabla 6.5 existen dos reglas que muestran que los alumnos que procrastinan en el acceso a los recursos externos pueden obtener un rendimiento alto. Estas reglas se repiten en los resultados obtenidos en los análisis de la segunda y tercera unidad, con una cobertura del 9,1 % y 6,1 % respectivamente. Analizando ambas reglas se puede ver que, además de la variable que indica procrastinación, la primera indica que los alumnos no procrastinan a la hora de realizar los cuestionarios, mientras que la segunda muestra que los alumnos acceden a muchos recursos. Este comportamiento parece indicar que la procrastinación en el acceso a recursos externos, por si sola, no permite relacionarse con un rendimiento bajo de los alumnos.

Analizando las reglas con mayor cobertura de cada uno de los instantes de análisis, es posible ver ciertas similitudes entre ellas. Un acceso alto a recursos internos combinado con la realización de tareas en instantes tempranos (variables `Timeto`) tiene como consecuencia una evaluación elevada. Por ejemplo, la regla `Resources` = `alto` $\wedge$ `Timetourls` = `bajo` entonces `Grade` = `alta` (Unidad 1) y la regla `Resources` = `alto` $\wedge$ `Timetoquizzes` = `bajo` entonces `Grade` = `alta` (Unidad 2) muestran esta similitud.

Podemos concluir en relación con la **Contribución 5** de esta tesis, que las reglas de asociación más influyentes para la asignatura dada ratifican que el tiempo que los alumnos emplean hasta la realización de acciones en el LMS influye en su evaluación final. Valores elevados en las variables de tiempo (procrastinación) están asociados a evaluaciones bajas, mientras que los accesos tempranos correlacionan con rendimientos elevados.

6.7. Relación con las contribuciones previas

El estudio de este capítulo se ha realizado para una única asignatura, en comparación con la investigación del Capítulos 4 y el Capítulo 5 que utilizan un conjunto elevado de asignaturas. Es por ello interesante ver en qué medida las conclusiones de ambos estudios son coherentes entre sí.

La Contribución 2 analiza la influencia de las variables del LMS en el rendimiento del alumno. Si bien las variables de evaluación son las más influyentes, existen otras que incluyen los accesos a recursos, entrada en el curso y visualización de tareas (Tabla 4.5) que incluyen en cierta medida la procrastinación del alumnado. Las principales diferencias entre ambos modelos radican en las diferencias en los conjuntos de datos utilizados. Este estudio se ha realizado para una única asignatura, en la que la plataforma de aprendizaje constituía un soporte importante para la misma, poseyendo recursos internos y externos, tareas evaluables a entregar y cuestionarios. Adicionalmente, se les requería a los alumnos de forma obligatoria que completasen todos los cuestionarios y entregasen al menos una tarea. Por el contrario, los estudios para múltiples asignaturas poseían multitud de estructuras, duraciones y contenidos distintos, siendo mucho más complejo medir la procrastinación del alumnado, aunque sí se exigía la evaluación de distintas tareas. Esta diferencia es la que hace que existan variaciones en el grado de influencia de las variables, aunque en ambos casos se detecta la correlación entre ellas.

La Contribución 4 está orientada a discutir la relación entre los grupos de alumnos y su rendimiento académico. En este caso, existen tres variables relacionadas con la procrastinación: `UAA`, `QAT` y `CIR`. Curiosamente, estas tres variables influyen directamente (sin ninguna otra variable) en el rendimiento del alumno, para todos los grupos que poseen correlación con la evaluación del alumnado. En todos los casos, la ausencia de procrastinación está asociada a una evaluación superior a la media y la procrastinación representa una evaluación por debajo de la media. Este comportamiento es por tanto igual al obtenido en el presente estudio asociado a la Contribución 5.

		Apriori			*Predictive Apriori*		
Antecedente	**Consecuente**	**Unidad 1**	**Unidad 2**	**Unidad 3**	**Unidad 1**	**Unidad 2**	**Unidad 3**
Resources = alto ∧ Timetoquizzes = bajo	Grade = alta		0,121		0,061	0,121	0,091
Resources = alto ∧ Timetoassigns = bajo	Grade = alta				0,061	0,061	0,061
Resources = alto ∧ Timetoresources = bajo	Grade = alta		0,121			0,121	0,091
Resources = alto ∧ Quizzes = bajo	Grade = alta			0,152		0,091	0,152
Urls = bajo ∧ Timetoquizzes = bajo	Grade = alta			0,121		0,091	0,121
Resources = bajo ∧ Timetoexternals = bajo	Grade = media				0,061		0,061
Resources = bajo ∧ Timetoresources = alto ∧ Timetoassigns = bajo	Grade = media				0,061	0,061	
Resources = alto ∧ Timetofirst = alto	Grade = baja					0,091	0,091
Urls = alto ∧ Quizzes = bajo	Grade = baja					0,091	0,091
Timetoquizzes = bajo ∧ Timetoexternals = alto	Grade = alta					0,091	0,091
Timetoquizzes = bajo ∧ Timetofirst = medio	Grade = alta					0,091	0,061
Resources = alto ∧ Timetourls = alto	Grade = alta					0,061	0,061
Resources = alto ∧ Timetofirst = bajo	Grade = alta					0,061	0,061

Tabla 6.5: Reglas de asociación obtenidas que se repiten en al menos dos unidades didácticas en los algoritmos seleccionados.

Capítulo 7

Conclusiones

La evaluación de los alumnos en asignaturas dentro de una plataforma de aprendizaje guarda una relación con las acciones que los estudiantes realizan en ella. Hemos creado modelos capaces de predecir dicha evaluación en instantes tempranos de la asignatura, utilizando tan solo los registros generados por el LMS a partir de las acciones de los alumnos. Los modelos detectan los alumnos en riesgo y suspensos con tan solo el 10 % de la duración de las asignaturas. A partir del 25 % también es posible identificar a los alumnos excelentes, creciendo su exactitud hasta un 93 % a mitad de la asignatura. Los algoritmos Árboles de Decisión y *MultiLayer Perceptron* ofrecen los mejores resultados, pudiendo seleccionar el más adecuado en función del coste de los falsos positivos y negativos que estime el profesor.

Las variables relacionadas con las acciones que realizan los alumnos en la plataforma de aprendizaje son muy influyentes a la hora de predecir el rendimiento en instantes tempranos. El acceso a los recursos que cuelgan los profesores tiene un alto impacto en la detección de los alumnos en riesgo de no superar la asignatura y aquellos suspensos. Adicionalmente, el acceso a la asignatura de forma reiterada al inicio del curso permite detectar a los alumnos excelentes. A medida que se tiene más información de los alumnos, las variables relacionadas con la evaluación obtenida por el alumno hasta el instante de predicción adquieren una mayor influencia en la predicción.

Las variables obtenidas de los registros (acciones y evaluación) para las etapas iniciales de las asignaturas permiten detectar patrones de acciones comunes de los estudiantes dentro del LMS. Mediante algoritmos de agrupamiento se han obtenido seis grupos diferentes de alumnos con patrones de acciones similares, repitiéndose los mismos grupos en todos los instantes tempranos de análisis estudiados.

De estos seis grupos identificados, cuatro guardan relación con la evaluación de sus estudiantes. Los patrones donde los alumnos no procrastinan a la hora de acceder a los recursos de la plataforma están relacionados con una evaluación superior. Por el contrario, aquellos grupos cuyos alumnos muestran una procrastinación más elevada están relacionados con un menor rendimiento.

Una característica importante es que todos los resultados mencionados anteriormente son válidos para un amplio conjunto de asignaturas heterogéneas. Esto implica que los resultados no dependen de la metodología de enseñanza de la asignatura, ni de su duración o disciplina.

Un estudio paralelo se ha centrado en una única asignatura, pero entendiendo como rendimiento del alumno su evaluación final en lugar de las calificaciones obtenidas en las tareas publicadas en el LMS. En este caso, la información es analizada mediante reglas que asociación que permiten relacionar las acciones que realizan los alumnos en el LMS con su evaluación final. Las reglas obtenidas muestran un comportamiento similar al estudio previo de agrupamiento, tanto tras la finalización de las diferentes unidades didácticas como al finalizar la asignatura: los alumnos que procrastinan obtienen una calificación final más baja que aquellos que no lo hacen.

Los resultados obtenidos de los diferentes estudios muestran cómo las acciones de los alumnos dentro del LMS son una información valiosa para predecir su rendimiento académico e identificar distintos patrones de comportamiento comunes. Este enfoque es válido tanto para modelos generalistas con información heterogénea de diferentes asignaturas como para modelos específicos basados en una única asignatura.

Capítulo 8

Trabajo Futuro

Esta tesis doctoral abre nuevas vías de trabajos de investigación a desarrollar en el futuro. En primer lugar, podrían incluirse los modelos creados dentro de la plataforma de aprendizaje. De esta forma, los profesores y los alumnos podrían recibir información o mensajes sobre el rendimiento académico en etapas iniciales del curso para poder tomar acciones de forma temprana. Con este enfoque se generarían nuevos indicadores o visualizaciones dentro de la plataforma que mostrarían la relación entre las acciones y el rendimiento esperado de los alumnos [75].

Los datos empleados en estos estudios están obtenidos únicamente de los *logs* de la plataforma de aprendizaje. De cara a enriquecer los modelos predictivos, parece interesante incluir información de otra naturaleza como datos sociodemográficos [76] o información de la interacción de los estudiantes con el LMS en cursos académicos anteriores.

Otra posible línea de trabajo incluye el análisis de la secuencia en el que los alumnos realizan las acciones y cómo este orden puede estar relacionado con la evaluación final del alumno. Dentro de este enfoque, algoritmos como las redes recurrentes como LSTM o GRU podrían proporcionar beneficios en la predicción [77]. Por ejemplo, se podría intentar prever si un alumno va a completar una actividad en función de la secuencia de sus acciones previas.

Otra posible línea de investigación incluye la utilización de otros algoritmos de aprendizaje automático para ver si éstos pueden obtener mayor precisión a la obtenida en esta investigación. Algoritmos como XGBoost [78, 79] han tenido un crecimiento en su uso en los últimos años y se han comenzado a aplicar dentro del ámbito del *e-learning*. Asociado al uso de estos algoritmos es necesario utilizar técnicas de explicabilidad como SHAP [80] o LIME [81] para poder analizar y entender las predicciones proporcionadas.

Los modelos generados en esta investigación son generalistas pero basados en una única plataforma de aprendizaje. Otro posible estudio sería realizar el mismo estudio sobre otra u otras plataformas de aprendizaje y comparar los resultados obtenidos. Dentro de este punto podrían incluirse otros LMSs utilizados por otras instituciones o incluso analizar el mismo comportamiento en cursos *online* masivos

y abiertos (*Massive Online Open Courses*, MOOCs).

Por último, la información masiva de registros (*logs*) de la que disponemos podría utilizarse para realizar agrupaciones de asignaturas. De este modo, se podrían analizar los distintos tipos de asignaturas en función de la interacción que los alumnos tienen con el LMS. La información extraída podría utilizarse para que los profesores supiesen en qué grupo se encuentra su asignatura, información sobre los otros tipos de asignaturas existentes, así como la correlación con una menor o mayor correlación con el rendimiento del alumnado. De este modo, se podrían tomar medidas para mejorar la implantación de asignaturas en un LMS en aras de incrementar el rendimiento del alumnado.

Apéndice A

Listado de variables de tipo acción

El siguiente listado muestra las diferentes variables de tipo acción generadas a partir de las acciones de los estudiantes dentro de la plataforma de aprendizaje. Esta lista se ha dividido por la tipología de la acción realizada (acceder a la asignatura, acceder a un recurso, etc.) para una mejor lectura. Para cada una de las acciones se indica el nombre y su descripción. Todas ellas indican porcentajes con valores comprendidos en el intervalo [0-100]

- Variables de acceso a la asignatura:
  - `CouseViewPct`: Porcentaje de accesos a la asignatura con respecto al total de accesos de todos los estudiantes.
  - `CourseViewTime{1,2,3,4,5}`: Instante dentro de la duración de la asignatura en el que se accede a la asignatura relativizado en base a su duración (porcentaje). Se generan 5 variables una para cada una de las primeras 5 acciones de este tipo.
  - `CourseViewTimePct`: Media geométrica de los valores generados en las variables anteriores.
- Variables de visualización de un recurso:
  - `ResourceViewPct`: Porcentaje de accesos a los recursos internos de la asignatura con respecto al total de accesos de todos los estudiantes.
  - `ResourceViewTime{1,2,3,4,5}`: Instante dentro de la duración de la asignatura en el que se accede a un recurso interno relativizado en base a su duración (porcentaje). Se generan 5 variables una para cada una de las primeras 5 acciones de este tipo.
  - `ResourceViewTimePct`: Media geométrica de los valores generados en las variables anteriores.
  - `ResourceViewUniquePct`: Porcentaje de recursos internos accedidos por el estudiante con respecto al número total de recursos únicos de la asignatura.
- Variables de visualización de un recurso externo:

  - `UrlViewPct`: Porcentaje de accesos a los recursos externos de la asignatura con respecto al total de accesos de todos los estudiantes.
  - `UrlViewTime{1,2,3,4,5}`: Instante dentro de la duración de la asignatura en el que se accede a un recurso externo relativizado en base a su duración (porcentaje). Se generan 5 variables una para cada una de las primeras 5 acciones de este tipo.
  - `UrlViewTimePct`: Media geométrica de los valores generados en las variables anteriores.
  - `UrlViewUniquePct`: Porcentaje de recursos externos accedidos por el estudiante con respecto al número total de recursos únicos de la asignatura.

- Variables de visualización de una tarea evaluable:

  - `AssignViewPct`: Porcentaje de accesos a las tareas de la asignatura con respecto al total de accesos de todos los estudiantes.
  - `AssignViewTime{1,2,3}`: Instante dentro de la duración de la asignatura en el que se accede a una tarea relativizado en base a su duración (porcentaje). Se generan 3 variables una para cada una de las primeras 3 acciones de este tipo.
  - `AssignViewTimePct`: Media geométrica de los valores generados en las variables anteriores.
  - `AssignViewUniquePct`: Porcentaje de tareas accedidas por el estudiante con respecto al número total de tareas únicas de la asignatura.

- Variables de visualización de un cuestionario:

  - `QuizViewPct`: Porcentaje de accesos a los cuestionarios de la asignatura con respecto al total de accesos de todos los estudiantes.
  - `QuizViewTime{1,2,3}`: Instante dentro de la duración de la asignatura en el que se accede a un cuestionario relativizado en base a su duración (porcentaje). Se generan 3 variables una para cada una de las primeras 3 acciones de este tipo.
  - `QuizViewTimePct`: Media geométrica de los valores generados en las variables anteriores.
  - `QuizViewUniquePct`: Porcentaje de cuestionarios accedidas por el estudiante con respecto al número total de cuestionarios únicos de la asignatura.

- Variables de envío de una tarea evaluable:

  - `AssignSubmitPct`: Porcentaje de envíos de las tareas de la asignatura con respecto al total de envíos de todos los estudiantes.
  - `AssignSubmitTime{1,2,3}`: Instante dentro de la duración de la asignatura en el que se envía una tarea relativizado en base a su duración (porcentaje). Se generan 3 variables una para cada una de las primeras 3 acciones de este tipo.
  - `AssignSubmitTimePct`: Media geométrica de los valores generados en las variables anteriores.
  - `AssignSubmitUniquePct`: Porcentaje de tareas enviadas por el estu-

diante con respecto al número total de tareas únicas de la asignatura.

- Variables de envío de comienzo de un cuestionario:
  - `QuizAttempPct`: Porcentaje de comienzo de intento de realización de cuestionarios de la asignatura con respecto al total de intentos de todos los estudiantes.
  - `QuizAttemptTime{1,2,3}`: Instante dentro de la duración de la asignatura en el que se comienza un cuestionario relativizado en base a su duración (porcentaje). Se generan 3 variables una para cada una de las primeras 3 acciones de este tipo.
  - `QuizAttemptTimePct`: Media geométrica de los valores generados en las variables anteriores.
  - `QuizAttemptUniquePct`: Porcentaje de cuestionarios comenzados por el estudiante con respecto al número total de cuestionarios únicos de la asignatura.
- Variables de envío de finalización de un cuestionario:
  - `QuizCloseAttempPct`: Porcentaje de envíos de los cuestionarios de la asignatura con respecto al total de envíos de todos los estudiantes.
  - `QuizCloseAttemptTime{1,2,3}`: Instante dentro de la duración de la asignatura en el que se envía un cuestionario relativizado en base a su duración (porcentaje). Se generan 3 variables una para cada una de las primeras 3 acciones de este tipo.
  - `QuizCloseAttemptTimePct`: Media geométrica de los valores generados en las variables anteriores.
  - `QuizCloseAttemptUniquePct`: Porcentaje de cuestionarios enviados por el estudiante con respecto al número total de cuestionarios únicos de la asignatura.
- Variables de visualización de foros:
  - `ForumViewForumPct`: Porcentaje de visitas al foro con respecto al total de visitas realizadas por todos los estudiantes.
  - `ForumViewDiscussionPct`: Porcentaje de visitas a discusiones en los foros con respecto al número total de visitas a discusiones de todos los estudiantes.

Apéndice B

Listado de variables de tipo evaluación

El siguiente listado muestra las diferentes variables de tipo evaluación generadas a partir de las evaluaciones obtenidas por los alumnos previo al instante de análisis y que están almacenadas dentro de los registros de la plataforma de aprendizaje.. Esta lista se ha dividido por la tipología de la tarea a evaluar (opcional u obligatoria) para una mejor lectura. Para cada una de las variables se indica el nombre, su tipología (entre corchetes) y su descripción.

- Variables de evaluación de tareas obligatorias:
  - `AccomplishMandatory` [0 ó 1]: indicador de si al menos una tarea obligatoria ha sido enviada.
  - `AccomplishMandatoryGrade` [0-10]: evaluación media de las tareas obligatorias del alumno usando las tareas enviadas hasta el instante seleccionado.
  - `AccomplishMandatoryPctGraded` [0-100]: porcentaje de tareas obligatorias enviadas antes del instante de tiempo de evaluación con respecto al total de tareas obligatorias de la asignatura.
  - `AccomplishMandatoryPercentileGrade` [0-100]: percentil correspondiente a la evaluación del estudiante en las tareas obligatorias hasta el instante de predicción, en relación con el resto de los alumnos de la misma asignatura.
- Variables de evaluación de tareas obligatorias:
  - `AccomplishOptional` [0 ó 1]: indicador de si al menos una tarea opcional ha sido enviada.
  - `AccomplishOptionalGrade` [0-10]: evaluación media de las tareas opcionales del alumno usando las tareas enviadas hasta el instante seleccionado.
  - `AccomplishOptionalPctGraded` [0-100]: porcentaje de tareas opcionales enviadas antes del instante de tiempo de evaluación con respecto al total de tareas opcionales de la asignatura.
  - `AccomplishOptionalPercentileGrade` [0-100]: percentil correspon-

diente a la evaluación del estudiante en las tareas opcionales hasta el instante de predicción, en relación con el resto de los alumnos de la misma asignatura.

Apéndice C

Detalle de la búsqueda de hiperparámetros

Este anexo contiene los hiperparámetros utilizados en la búsqueda del mejor modelo predictivo para cada uno de los algoritmos utilizados. Las siguientes secciones enumeran los parámetros utilizados durante el entrenamiento de los modelos, utilizando los proporcionados por defecto por la librería `scikit-learn` [46] para el resto. Por último, también se indica para cada uno de los entrenamientos realizados cuáles son los mejores hiperparámetros encontrados.

C.1. Hiperparámetros del algoritmo Naïve Bayes

La Tabla C.1 muestra los hiperparámetros utilizados, así como los distintos valores probados, para la clase `GaussianNB` de `scikit-learn`.

Parámetro	Posibles valores
`var_smoothing`	1e-08, 1e-09, 1e-10

Tabla C.1: Opciones para la selección de hiperparámetros en el algoritmo Naïve-Bayes.

Los valores de los hiperparámetros con los mejores resultados fueron los siguientes (por umbral de corte e instante de predicción):

- Umbral de corte 2,5, 5,0 y 8,5:
  - Instantes de predicción 10 %, 25 %, 33 % y 50 %: `var_smoothing` = 1e-09.

C.2. Hiperparámetros del algoritmo Árboles de Decisión

La Tabla C.2 muestra los hiperparámetros utilizados, así como los distintos valores probados, para la clase `DecissionTreeClassifier` de `scikit-learn`.

Parámetro	Posibles valores
`criterion`	'gini', 'entropy'
`sppliter`	'best', 'normal'
`max_depth`	`None`, 5, 10, 15
`max_features`	`None`, 'auto', 'sqrt', 'log2'
`class_weight`	`None`, 'balanced'
`presort`	`True`, `False`

Tabla C.2: Opciones para la selección de hiperparámetros en el algoritmo Árboles de Decisión.

Los valores de los hiperparámetros con los mejores resultados fueron los siguientes (por umbral de corte e instante de predicción):

- Umbral de corte 2,5:
  - Instantes de predicción 10 %, 25 %, 33 % y 50 %: `splitter` = `random`, `presort` = `False`, `max_features` = `None`, `max_depth` = 10, `criterion` = `gini`, `class_weight` = `None`.
- Umbral de corte 5,0:
  - Instante de predicción 10 %: `splitter` = `best`, `presort` = `False`, `max_features` = `sqrt`, `max_depth` = 10, `criterion` = `gini`, `class_weight` = `None`.
  - Instantes de predicción 25 % y 50 %: `splitter` = `random`, `presort` = `True`, `max_features` = `None`, `max_depth` = 10, `criterion` = `gini`, `class_weight` = `balanced`.
  - Instante de predicción 33 %: `splitter` = `random`, `presort` = `True`, `max_features` = `None`, `max_depth` = 10, `criterion` = `entropy`, `class_weight` = `balanced`.
- Umbral de corte 8,5:
  - Instantes de predicción 10 %, 25 %, 33 % y 50 %: `splitter` = `best`, `presort` = `True`, `max_features` = `None`, `max_depth` = 5, `criterion` = `entropy`, `class_weight` = `None`.

C.3. Hiperparámetros del algoritmo Regresión Logística

La Tabla C.3 muestra los hiperparámetros utilizados, así como los distintos valores probados, para la clase `LogisticRegression` de `scikit-learn`.

Parámetro	Posibles valores
`penalty`	'l1', 'l2'
`tol`	1e-2, 1e-3, 1e-4, 1e-5
`solver`	'liblinear'
`max_iter`	50, 100, 200

Tabla C.3: Opciones para la selección de hiperparámetros en el algoritmo Regresión Logística.

Los valores de los hiperparámetros con los mejores resultados fueron los siguientes (por umbral de corte e instante de predicción):

- Umbral de corte 2,5:
  - Instantes de predicción 10 % y 25 %: `tol` = 0.0001, `solver` = `liblinear`, `penalty` = `l1`, `max_iter` = 200.
  - Instante de predicción 33 %: `tol` = 0,001, `solver` = `liblinear`, `penalty` = `l2`, `max_iter` = 50.
  - Instante de predicción 50 %: `tol` = 0,01, `solver` = `liblinear`, `penalty` = `l2`, `max_iter` = 100.
- Umbral de corte 5,0:
  - Instante de predicción 10 %: `tol` = 1e-05, `solver` = `liblinear`, `penalty` = `l2`, `max_iter` = 200.
  - Instantes de predicción 25 % y 33 %: `tol` = 0,001, `solver` = `liblinear`, `penalty` = `l1`, `max_iter` = 100.
  - Instante de predicción 50 %: `tol` = 1e-05, `solver` = `liblinear`, `penalty` = `l1`, `max_iter` = 200.
- Umbral de corte 8,5:
  - Instante de predicción 10 %: `tol` = 1e-05, `solver` = `liblinear`, `penalty` = `l2`, `max_iter` = 50.
  - Instante de predicción 25 %: `tol` = 0,01, `solver` = `liblinear`, `penalty` = `l1`, `max_iter` = 100.
  - Instante de predicción 33 %: `tol` = 0,0001, `solver` = `liblinear`, `penalty` = `l1`, `max_iter` = 100.
  - Instante de predicción 50 %: `tol` = 0,01, `solver` = `liblinear`, `penalty` = `l2`, `max_iter` = 100.

C.4. Hiperparámetros del algoritmo SVM

La Tabla C.4 muestra los hiperparámetros utilizados, así como los distintos valores probados, para la clase `SVC` de `scikit-learn`.

Parámetro	Posibles valores
`C`	1
`kernel`	'rbf'
`gamma`	'scale'
`tol`	1e-2, 1e-3, 1e-4
`probability`	`True`
`cache_size`	1024*4

Tabla C.4: Opciones para la selección de hiperparámetros en el algoritmo SVM.

Los valores de los hiperparámetros con los mejores resultados fueron los siguientes (por umbral de corte e instante de predicción):

- Umbrales de corte 2,5, 5,0 y 8,5.
  - Instantes de predicción 10 %, 25 %, 33 % y 50 %: `tol` = 0,01, `probability` = `True`, `kernel` = `rbf`, `gamma` = `scale`, `cache_size` = 4096, `C` = 1.

C.5. Hiperparámetros del algoritmo MultiLayer Perceptron

La Tabla C.5 muestra los hiperparámetros utilizados, así como los distintos valores probados, para la clase `MLPClassifier` de `scikit-learn`.

Parámetro	Posibles valores
`hidden_layers`	20, [20, 20]
`activation`	'identity', 'relu', 'tanh'
`solver`	'adam', 'sgd', 'lbfgs'
`alpha`	1, 0,1, 0,01, 0,001
`solver`	'constant', 'invscaling', 'adaptative'

Tabla C.5: Opciones para la selección de hiperparámetros en el algoritmo MultiLayer Perceptron.

Los valores de los hiperparámetros con los mejores resultados fueron los siguientes (por umbral de corte e instante de predicción):

- Umbral de corte 2,5:
  - Instantes de predicción 10 %, 25 %, 33 % y 50 %: `solver` = `lbfgs`, `learning_rate` = `constant`, `hidden_layer_sizes` = 20, `alpha` = 0,1, `activation` = `tanh`.
- Umbral de corte 5,0:
  - Instantes de predicción 10 % y 50 %: `solver` = `lbfgs`, `learning_rate` = `adaptive`, `hidden_layer_sizes` = 20, `alpha` = 0,01, `activation` = `relu`.
  - Instante de predicción 25 %: `solver` = `adam`, `learning_rate` = `constant`, `hidden_layer_sizes` = 20, `alpha` = 0,001, `activation` = `relu`.
  - Instante de predicción 33 %: `solver` = `lbfgs`, `learning_rate` = `invscaling`, `hidden_layer_sizes` = 20, `alpha` = 0,1, `activation` = `relu`.
- Umbral de corte 8,5:
  - Instante de predicción 10 %: `solver` = `adam`, `learning_rate` = `adaptive`, `hidden_layer_sizes` = 20, `alpha` = 0,001, `activation` = `relu`.
  - Instante de predicción 25 %: `solver` = `lbfgs`, `learning_rate` = `adaptive`, `hidden_layer_sizes` = 20, `alpha` = 1, `activation` = `relu`.
  - Instantes de predicción 33 % and 50 %: `solver` = `lbfgs`, `learning_rate` = `invscaling`, `hidden_layer_sizes` = 20, `alpha` = 1, `activation` = `relu`.

Apéndice D

Agregación de variables

Durante la ejecución de los algoritmos de agrupamiento se realizó una reducción de la dimensionalidad del conjunto de datos inicial. Para ello se utilizó el algoritmo `FeatureAgglomeration` proporcionado por la librería `scikit-learn` [46]. El conjunto inicial de variables fue reducido a las siguientes variables:

- `UAA`(*URL and assignment access*): Esta variable agrega las variables originales relacionadas con los instantes de tiempo en los que se consultan recursos externos y en los que se visualizan y entregan tareas.
- `MOA`(*Mandatory and Optional Assignment Evaluation*): Agregación de todas las variables relacionadas con las evaluaciones de los alumnos, tanto obligatorias como opcionales.
- `QAT`(*Quizzes Access Time*): Agregación de las variables relacionadas con los instantes de acceso, comienzo y finalización de los cuestionarios.
- `CIR`(*Course and Resource View*): Esta dimensión agrega las variables relacionadas con el instante de acceso a las asignaturas, así como a los recursos internos proporcionados.

La Tabla D.1 muestra la correspondencia entre las variables originales descritas en el Anexo A y en el Anexo B y las nuevas variables generadas.

URL and assignment access (UAA)	*Mandatory and optional assignment evaluation* (MOA)	*Quiz access time* (QAT)	*Course and resource view* (CIR)
UrlViewTime1	AccomplishMandatory	QuizViewTime1	CourseViewtime1
UrlViewTime2	AccomplishMandatoryGrade	QuizViewTime2	CourseViewtime2
UrlViewTime3	AccomplishMandatoryPctGraded	QuizViewTime3	CourseViewtime3
UrlViewTime4	AccomplishMandatoryPercentileGrade	QuizViewTimePct	CourseViewtime4
UrlViewTime5	AccomplishOptional	QuizAttemptTime1	CourseViewtime5
UrlViewTimePct	AccomplishOptional	QuizAttemptTime1	CourseViewtime5
AssignViewTime1	AccomplishOptionalGrade	QuizAttemptTime2	CourseViewtimePct
AssignViewTime2	AccomplishOptionalPctGraded	QuizAttemptTime3	ResourceViewTime1
AssignViewTime3	AccomplishOptionalPercentileGrade	QuizAttemptTimePct	ResourceViewTime2
AssignViewTimePct	CourseViewPct	QuizCloseAttemptTime1	ResourceViewTime3
AssignSubmitTime1	ResourceViewPct	QuizCloseAttemptTime2	ResourceViewTime4
AssignSubmitTime2	ResourceViewUniquePct	QuizCloseAttemptTime3	ResourceViewTime5
AssignSubmitTime3	UrlViewPct	QuizCloseAttemptTimePct	ResourceViewTimePct
AssignSubmitTimePct	UrlViewUniquePct		
	AssignViewPct		
	AssignViewUniquePct		
	QuizViewPct		
	QuizViewUniquePct		
	AssignSubmitPct		
	AssignSubmitUniquePct		
	QuizAttemptPct		
	QuizAttemptUniquePct		
	QuizCloseUniquePct		
	QuizCloseAttemptUniquePct		
	ForumViewForumPct		
	ForumViewDiscussionPct		

Tabla D.1: Correspondencia entre las variables originales y las variables generadas en los algoritmos de agrupamiento

Apéndice E

Reglas de asociación

Durante el análisis de la relación entre la procrastinación y la evaluación de los estudiantes, se han obtenido un conjunto de reglas que se adjuntan a continuación, agrupadas por el algoritmo utilizado y el instante de análisis al que pertenecen.

E.1. Reglas de asociación obtenidas con el algoritmo *Apriori*

Esta sección muestra las reglas obtenidas con el algoritmo *Apriori* en los distintos instantes de análisis de la asignatura. La Tabla E.1 muestra las 12 reglas obtenidas en el análisis tras la unidad didáctica 1.

Antecedente	**Consecuente**	**Cobertura**
`Quizzes` = `medio` ∧ `Timetoquizzes` = `medio`	`Grade` = `alta`	0,182
`Resources` = `alto` ∧ `Timetourls` = `bajo`	`Grade` = `alta`	0,152
`Resources` = `medio` ∧ `Quizzes` = `medio`	`Grade` = `alta`	0,121
`Resources` = `alto` ∧ `Timetofirst` = `medio`	`Grade` = `alta`	0,121
`Quizzes` = `medio` ∧ `Timetoresources` = `medio`	`Grade` = `alta`	0,121
`Resources` = `alto` ∧ `Urls` = `alto` ∧ `Timetourls` = `bajo`	`Grade` = `alta`	0,121
`Resources` = `alto` ∧ `Timetoquizzes` = `medio` ∧ `Timetourls` = `bajo`	`Grade` = `alta`	0,121
`Resources` = `alto` ∧ `Timetourls` = `bajo` ∧ `Timetofirst` = `medio`	`Grade` = `alta`	0,121
`Quizzes` = `medio` ∧ `Timetoquizzes` = `medio` ∧ `Timetourls` = `bajo`	`Grade` = `alta`	0,121
`Quizzes` = `medio` ∧ `Timetoquizzes` = `medio` ∧ `Timetoassignments` = `medio`	`Grade` = `alta`	0,121
`Timetoquizzes` = `bajo` ∧ `Timetoresources` = `bajo` ∧ `Timetoassignments` = `bajo`	`Grade` = `alta`	0,121
`Timetoquizzes` = `bajo` ∧ `Timetoresources` = `bajo` ∧ `Timetoassignments` = `bajo` ∧ `Timetofirst` = `bajo`	`Grade` = `alta`	0,121

Tabla E.1: Reglas obtenidas con el algoritmo *Apriori* al terminar la primera unidad didáctica.

La Tabla E.2 y la Tabla E.3 se corresponden con el listado de reglas obtenidas

al finalizar la segunda y la tercera unidad didáctica respectivamente.

Antecedente	**Consecuente**	**Cobertura**
Timetoresources = medio ∧ Timetourls = medio	Grade = baja	0,152
Resources = medio ∧ Timetourls = bajo	Grade = media	0,121
Resources = alto ∧ Timetoquizzes = bajo	Grade = alta	0,121
Resources = alto ∧ Timetoresources = bajo	Grade = alta	0,121
Timetoresources = medio ∧ Timetourls = bajo	Grade = media	0,121
Timetoresources = medio ∧ Timetourls = medio ∧ Timetoassignments = alto	Grade = baja	0,121

Tabla E.2: Reglas obtenidas con el algoritmo *Apriori* al terminar la segunda unidad didáctica.

Antecedente	**Consecuente**	**Cobertura**
Quizzes = bajo ∧ Timetoassignments = alto	Grade = baja	0,182
Resources = alto ∧ Quizzes = bajo	Grade = baja	0,152
Quizzes = bajo ∧ Timetoresources = alto	Grade = baja	0,152
Quizzes = bajo ∧ Timetofirst = alto	Grade = baja	0,152
Timetoquizzes = alto ∧ Timetofirst = alto	Grade = baja	0,152
Resources = alto ∧ Quizzes = bajo ∧ Timetoquizzes = alto	Grade = baja	0,152
Quizzes = bajo ∧ Timetoquizzes = alto ∧ Timetoassignments = alto	Grade = baja	0,152
Resources = bajo ∧ Quizzes = bajo	Grade = baja	0,121
Resources = bajo ∧ Timetoassignments = alto	Grade = baja	0,121
Urls = bajo ∧ Quizzes = bajo	Grade = baja	0,121
Urls = bajo ∧ Timetoassignments = alto	Grade = baja	0,121
Urls = bajo ∧ Timetoquizzes = bajo	Grade = alta	0,121
Quizzes = bajo ∧ Timetourls = alto	Grade = baja	0,121
Timetoquizzes = alto ∧ Timetoresources = alto	Grade = baja	0,121
Timetoquizzes = alto ∧ Timetoassignments = medio	Grade = baja	0,121
Resources = bajo ∧ Quizzes = bajo ∧ Timetourls = alto	Grade = baja	0,121
Quizzes = bajo ∧ Timetoquizzes = alto ∧ Timetoresources = alto	Grade = baja	0,121
Quizzes = bajo ∧ Timetoquizzes = alto ∧ Timetofirst = alto	Grade = baja	0,121
Quizzes = bajo ∧ Timetoassignments = alto ∧ Timetofirst = alto	Grade = baja	0,121

Tabla E.3: Reglas obtenidas con el algoritmo *Apriori* al terminar la tercera unidad didáctica.

Por último, la Tabla E.4 muestra las reglas obtenidas en el análisis con las acciones al finalizar la asignatura.

E.2. Reglas de asociación obtenidas con el algoritmo *Predictive Apriori*

Esta sección muestra las reglas obtenidas con el algoritmo *Predictive Apriori* en los distintos instantes de análisis de la asignatura. La Tabla E.5 muestra las 47 reglas obtenidas en el análisis tras la unidad didáctica 1.

Antecedente	**Consecuente**	**Cobertura**
Quizzes = medio ∧ Timetofirst = alto	Grade = baja	0,152
Urls = bajo ∧ Quizzes = bajo	Grade = baja	0,121
Urls = bajo ∧ Timetoassignments = alto	Grade = baja	0,121
Timetoresources = alto ∧ Timetoassignments = alto	Grade = baja	0,121
Quizzes = alto ∧ Timetoresources = bajo ∧ Timetoassignments = bajo	Grade = alta	0,121
Quizzes = alto ∧ Timetoassignments = bajo ∧ Timetofirst = bajo	Grade = alta	0,121
Quizzes = alto ∧ Timetoresources = bajo ∧ Timetoassignments = bajo ∧ Timetofirst = bajo	Grade = alta	0,121

Tabla E.4: Reglas obtenidas con el algoritmo *Apriori* al terminar la asignatura.

La Tabla E.6 y la Tabla E.7 se corresponden con el listado de reglas obtenidas al finalizar la segunda y la tercera unidad didáctica respectivamente.

Por último, la Tabla E.8 muestra las reglas obtenidas en el análisis con las acciones al finalizar la asignatura.

Antecedente	**Consecuente**	**Cobertura**	***Accuracy***
Quizzes = medio ∧ Timetoquizzes = medio	Grade = alto	0,182	0,99194
Resources = alto ∧ Timetourls = bajo	Grade = alto	0,152	0,99028
Resources = medio ∧ Quizzes = medio	Grade = alto	0,121	0,98729
Resources = alto ∧ Timetofirst = medio	Grade = alto	0,121	0,98729
Quizzes = medio ∧ Timetoresources = medio	Grade = alto	0,121	0,98729
Timetoquizzes = bajo ∧ Timetoresources = bajo ∧ Timetoassignments = bajo	Grade = alto	0,121	0,98729
Resources = medio ∧ Quizzes = bajo	Grade = medio	0,091	0,98168
Resources = medio ∧ Timetoassignments = bajo	Grade = alto	0,091	0,98168
Resources = medio ∧ Timetofirst = bajo	Grade = alto	0,091	0,98168
Resources = alto ∧ Quizzes = medio	Grade = alto	0,091	0,98168
Urls = medio ∧ Quizzes = medio	Grade = alto	0,091	0,98168
Quizzes = medio ∧ Timetoresources = bajo	Grade = alto	0,091	0,98168
Timetoquizzes = medio ∧ Timetofirst = medio	Grade = alto	0,091	0,98168
Timetoquizzes = alto ∧ Timetoresources = medio	Grade = medio	0,091	0,98168
Urls = bajo ∧ Timetoquizzes = bajo ∧ Timetoassignments = bajo	Grade = alto	0,091	0,98168
Urls = bajo ∧ Timetoresources = bajo ∧ Timetoassignments = bajo	Grade = alto	0,091	0,98168
Urls = bajo ∧ Timetoassignments = bajo ∧ Timetofirst = bajo	Grade = alto	0,091	0,98168
Urls = alto ∧ Timetoquizzes = medio ∧ Timetourls = bajo	Grade = alto	0,091	0,98168
Quizzes = bajo ∧ Timetoquizzes = alto ∧ Timetourls = medio	Grade = medio	0,091	0,98168
Timetoquizzes = bajo ∧ Timetourls = medio ∧ Timetoassignments = bajo ∧ Timetofirst = bajo	Grade = alto	0,091	0,98168
Resources = bajo ∧ Quizzes = alto	Grade = alta	0,061	0,97058
Resources = bajo ∧ Timetourls = bajo	Grade = media	0,061	0,97058
Resources = medio ∧ Timetoquizzes = medio	Grade = alta	0,061	0,97058
Resources = alto ∧ Timetoquizzes = bajo	Grade = alta	0,061	0,97058
Resources = alto ∧ Timetourls = medio	Grade = media	0,061	0,97058
Resources = alto ∧ Timetoassignments = bajo	Grade = alta	0,061	0,97058
Urls = bajo ∧ Timetoquizzes = medio	Grade = alta	0,061	0,97058
Urls = alto ∧ Timetoquizzes = alto	Grade = media	0,061	0,97058
Urls = alto ∧ Timetourls = medio	Grade = media	0,061	0,97058
Quizzes = medio ∧ Timetourls = alto	Grade = alta	0,061	0,97058
Quizzes = alto ∧ Timetoresources = alto	Grade = alta	0,061	0,97058
Quizzes = alto ∧ Timetourls = bajo	Grade = alta	0,061	0,97058
Quizzes = alto ∧ Timetoassignments = bajo	Grade = alta	0,061	0,97058
Timetoquizzes = alto ∧ Timetourls = bajo	Grade = media	0,061	0,97058
Timetoresources = medio ∧ Timetoassignments = bajo	Grade = alta	0,061	0,97058
Timetoassignments = medio ∧ Timetofirst = alto	Grade = alta	0,061	0,97058
Resources = bajo ∧ Quizzes = bajo ∧ Timetoquizzes = alto	Grade = media	0,061	0,97058
Resources = bajo ∧ Quizzes = bajo ∧ Timetoresources = alto	Grade = media	0,061	0,97058
Resources = bajo ∧ Timetoquizzes = bajo ∧ Timetoassignments = medio	Grade = baja	0,061	0,97058
Resources = bajo ∧ Timetoquizzes = medio ∧ Timetourls = medio	Grade = alta	0,061	0,97058
Resources = bajo ∧ Timetoquizzes = medio ∧ Timetofirst = alto	Grade = alta	0,061	0,97058
Resources = bajo ∧ Timetoresources = alto ∧ Timetoassignments = bajo	Grade = media	0,061	0,97058
Resources = medio ∧ Quizzes = alto ∧ Timetofirst = medio	Grade = bajo	0,061	0,97058
Resources = medio ∧ Timetoresources = medio ∧ Timetourls = bajo	Grade = media	0,061	0,97058
Resources = alto ∧ Timetoresources = medio ∧ Timetoassignments = medio	Grade = alta	0,061	0,97058
Resources = alto ∧ Timetourls = alto ∧ Timetoassignments = alto	Grade = baja	0,061	0,97058
Urls = alto ∧ Quizzes = medio ∧ Timetoassignments = medio	Grade = alta	0,061	0,97058

Tabla E.5: Reglas obtenidas con el algoritmo *Predictive Apriori* al terminar la primera unidad didáctica.

Antecedente	**Consecuente**	**Cobertura**	***Accuracy***
Timetoresources = medio Timetourls = medio	Grade = baja	0,152	0,99001
Resources = medio Timetourls = bajo	Grade = media	0,121	0,98725
Resources = alto Timetoquizzes = bajo	Grade = alta	0,121	0,98725
Resources = alto Timetoresources = bajo	Grade = alta	0,121	0,98725
Timetoresources = medio ∧ Timetourls = bajo	Grade = media	0,121	0,98725
Resources = alto ∧ Quizzes = bajo	Grade = baja	0,091	0,98224
Resources = alto ∧ Timetourls = medio	Grade = baja	0,091	0,98224
Resources = alto ∧ Timetofirst = alto	Grade = baja	0,091	0,98224
Urls = bajo ∧ Timetoquizzes = bajo	Grade = alta	0,091	0,98224
Urls = alto ∧ Quizzes = bajo	Grade = baja	0,091	0,98224
Urls = alto ∧ Timetourls = medio	Grade = baja	0,091	0,98224
Timetoquizzes = bajo ∧ Timetourls = alto	Grade = alta	0,091	0,98224
Timetoquizzes = bajo ∧ Timetofirst = medio	Grade = alta	0,091	0,98224
Timetourls = medio ∧ Timetofirst = medio	Grade = baja	0,091	0,98224
Resources = bajo ∧ Quizzes = medio ∧ Timetoquizzes = bajo	Grade = alta	0,091	0,98224
Urls = alto ∧ Timetoquizzes = alto ∧ Timetofirst = alto	Grade = baja	0,091	0,98224
Quizzes = bajo ∧ Timetoquizzes = alto ∧ Timetoassignments = medio	Grade = baja	0,091	0,98224
Quizzes = bajo ∧ Timetoresources = alto ∧ Timetoassignments = medio	Grade = baja	0,091	0,98224
Quizzes = bajo ∧ Timetoassignments = medio ∧ Timetofirst = alto	Grade = baja	0,091	0,98224
Quizzes = medio ∧ Timetoquizzes = bajo ∧ Timetoresources = bajo	Grade = alta	0,091	0,98224
Quizzes = medio ∧ Timetoresources = bajo ∧ Timetoassignments = bajo	Grade = alta	0,091	0,98224
Resources = bajo ∧ Urls = medio	Grade = media	0,061	0,97262
Resources = medio ∧ Quizzes = medio	Grade = media	0,061	0,97262
Resources = alto ∧ Timetourls = alto	Grade = alta	0,061	0,97262
Resources = alto ∧ Timetoassignments = bajo	Grade = alta	0,061	0,97262
Resources = alto ∧ Timetofirst = bajo	Grade = alta	0,061	0,97262
Urls = alto ∧ Timetoquizzes = bajo	Grade = alta	0,061	0,97262
Urls = alto ∧ Timetoresources = bajo	Grade = alta	0,061	0,97262
Urls = alto ∧ Timetoassignments = bajo	Grade = alta	0,061	0,97262
Quizzes = bajo ∧ Timetoassignments = bajo	Grade = media	0,061	0,97262
Quizzes = bajo ∧ Timetofirst = bajo	Grade = media	0,061	0,97262
Timetoquizzes = medio ∧ Timetoassignments = bajo	Grade = media	0,061	0,97262
Timetoresources = bajo ∧ Timetofirst = medio	Grade = alta	0,061	0,97262
Timetoresources = alto ∧ Timetofirst = bajo	Grade = media	0,061	0,97262
Timetourls = bajo ∧ Timetoassignments = alto	Grade = media	0,061	0,97262
Timetourls = medio ∧ Timetoassignments = bajo	Grade = alta	0,061	0,97262
Timetoassignments = bajo ∧ Timetofirst = medio	Grade = alta	0,061	0,97262
Resources = bajo ∧ Quizzes = bajo ∧ Timetoassignments = alto	Grade = media	0,061	0,97262
Resources = bajo ∧ Timetoresources = alto ∧ Timetoassignments = bajo	Grade = media	0,061	0,97262
Resources = bajo ∧ Timetoresources = alto ∧ Timetoassignments = alto	Grade = media	0,061	0,97262
Resources = bajo ∧ Timetoassignments = alto ∧ Timetofirst = alto	Grade = media	0,061	0,97262
Resources = medio ∧ Urls = alto ∧ Quizzes = alto	Grade = media	0,061	0,97262
Resources = medio ∧ Urls = alto ∧ Timetoassignments = alto	Grade = media	0,061	0,97262
Resources = medio ∧ Quizzes = bajo ∧ Timetourls = medio	Grade = baja	0,061	0,97262
Resources = medio ∧ Timetoquizzes = medio ∧ Timetoresources = bajo	Grade = media	0,061	0,97262
Resources = medio ∧ Timetoquizzes = medio ∧ Timetofirst = bajo	Grade = media	0,061	0,97262
Resources = medio ∧ Timetourls = medio ∧ Timetofirst = alto	Grade = baja	0,061	0,97262
Resources = alto ∧ Urls = alto ∧ Timetoassignments = alto	Grade = baja	0,061	0,97262
Resources = alto ∧ Quizzes = alto ∧ Timetoassignments = medio	Grade = alta	0,061	0,97262
Resources = alto ∧ Timetoquizzes = alto ∧ Timetoresources = alto	Grade = baja	0,061	0,97262
Resources = alto ∧ Timetoresources = medio ∧ Timetoassignments = alto	Grade = baja	0,061	0,97262
Urls = alto ∧ Timetoassignments = medio ∧ Timetofirst = alto	Grade = baja	0,061	0,97262

Tabla E.6: Reglas obtenidas con el algoritmo *Predictive Apriori* al terminar la segunda unidad didáctica.

Antecedente	Consecuente	Cobertura	*Accuracy*
Quizzes = bajo Timetoassignments = alto	Grade = baja	0,182	0,99129
Resources = alto Quizzes = bajo	Grade = baja	0,152	0,98933
Quizzes = bajo Timetoresources = alto	Grade = baja	0,152	0,98933
Quizzes = bajo Timetofirst = alto	Grade = baja	0,152	0,98933
Timetoquizzes = alto Timetofirst = alto	Grade = baja	0,152	0,98933
Resources = bajo ∧ Quizzes = bajo	Grade = baja	0,121	0,98588
Resources = bajo ∧ Timetoassignments = alto	Grade = baja	0,121	0,98588
Urls = bajo ∧ Quizzes = bajo	Grade = baja	0,121	0,98588
Urls = bajo ∧ Timetoquizzes = bajo	Grade = alta	0,121	0,98588
Urls = bajo ∧ Timetoassignments = alto	Grade = baja	0,121	0,98588
Quizzes = bajo ∧ Timetourls = alto	Grade = baja	0,121	0,98588
Timetoquizzes = alto ∧ Timetoresources = alto	Grade = baja	0,121	0,98588
Timetoquizzes = alto ∧ Timetoassignments = medio	Grade = baja	0,121	0,98588
Resources = bajo ∧ Timetoquizzes = alto	Grade = baja	0,091	0,97953
Resources = alto ∧ Timetoquizzes = bajo	Grade = alta	0,091	0,97953
Resources = alto ∧ Timetoresources = bajo	Grade = alta	0,091	0,97953
Resources = alto ∧ Timetofirst = alto	Grade = baja	0,091	0,97953
Urls = bajo ∧ Timetoquizzes = alto	Grade = baja	0,091	0,97953
Urls = bajo ∧ Timetoassignments = bajo	Grade = alta	0,091	0,97953
Urls = medio ∧ Timetofirst = alto	Grade = baja	0,091	0,97953
Urls = alto ∧ Quizzes = bajo	Grade = baja	0,091	0,97953
Timetoquizzes = bajo ∧ Timetourls = alto	Grade = alta	0,091	0,97953
Timetoquizzes = alto ∧ Timetourls = alto	Grade = baja	0,091	0,97953
Urls = bajo ∧ Quizzes = medio ∧ Timetourls = alto	Grade = alta	0,091	0,97953
Quizzes = bajo ∧ Timetoquizzes = alto ∧ Timetoresources = medio	Grade = baja	0,091	0,97953
Quizzes = bajo ∧ Timetoquizzes = alto ∧ Timetofirst = medio	Grade = baja	0,091	0,97953
Resources = bajo ∧ Timetoresources = medio	Grade = baja	0,061	0,96736
Resources = bajo ∧ Timetourls = bajo	Grade = media	0,061	0,96736
Resources = bajo ∧ Timetofirst = medio	Grade = baja	0,061	0,96736
Resources = medio ∧ Urls = bajo	Grade = alta	0,061	0,96736
Resources = medio ∧ Timetoassignments = alto	Grade = media	0,061	0,96736
Resources = alto ∧ Timetourls = alto	Grade = alta	0,061	0,96736
Resources = alto ∧ Timetoassignments = bajo	Grade = alta	0,061	0,96736
Resources = alto ∧ Timetofirst = bajo	Grade = alta	0,061	0,96736
Urls = alto ∧ Quizzes = medio	Grade = media	0,061	0,96736
Quizzes = medio ∧ Timetourls = bajo	Grade = media	0,061	0,96736
Timetoquizzes = bajo ∧ Timetofirst = medio	Grade = alta	0,061	0,96736
Timetoquizzes = medio ∧ Timetoresources = bajo	Grade = alta	0,061	0,96736
Timetoquizzes = alto ∧ Timetourls = bajo	Grade = baja	0,061	0,96736
Resources = bajo ∧ Urls = medio ∧ Timetoquizzes = bajo	Grade = media	0,061	0,96736
Resources = bajo ∧ Urls = medio ∧ Timetofirst = bajo	Grade = media	0,061	0,96736
Resources = bajo ∧ Quizzes = alto ∧ Timetoassignments = medio	Grade = media	0,061	0,96736
Resources = medio ∧ Urls = alto ∧ Timetoquizzes = bajo	Grade = media	0,061	0,96736
Resources = medio ∧ Urls = alto ∧ Timetofirst = alto	Grade = media	0,061	0,96736
Resources = medio ∧ Timetoquizzes = bajo ∧ Timetourls = bajo	Grade = media	0,061	0,96736
Resources = alto ∧ Quizzes = alto ∧ Timetoresources = medio	Grade = media	0,061	0,96736
Resources = alto ∧ Timetoresources = alto ∧ Timetourls = medio	Grade = baja	0,061	0,96736
Resources = alto ∧ Timetoresources = alto ∧ Timetoassignments = alto	Grade = baja	0,061	0,96736
Urls = bajo ∧ Timetoresources = alto ∧ Timetofirst = medio	Grade = baja	0,061	0,96736
Urls = alto ∧ Quizzes = alto ∧ Timetoresources = medio	Grade = media	0,061	0,96736
Quizzes = medio ∧ Timetoquizzes = bajo ∧ Timetoresources = alto	Grade = media	0,061	0,96736
Resources = medio ∧ Urls = medio ∧ Quizzes = medio ∧ Timetoresources = medio	Grade = baja	0,061	0,96736
Urls = bajo ∧ Quizzes = alto ∧ Timetoresources = bajo ∧ Timetofirst = bajo	Grade = alta	0,061	0,96736

Tabla E.7: Reglas obtenidas con el algoritmo *Predictive Apriori* al terminar la tercera unidad didáctica.

Antecedente	**Consecuente**	**Cobertura**	***Accuracy***
Quizzes = medio ∧ Timetofirst = alto	Grade = baja	0,152	0,99109
Urls = bajo ∧ Quizzes = bajo	Grade = baja	0,121	0,98884
Urls = bajo ∧ Timetoassignments = alto	Grade = baja	0,121	0,98884
Timetoresources = alto ∧ Timetoassignments = alto	Grade = baja	0,121	0,98884
Quizzes = alto ∧ Timetoresources = bajo Timetoassignments = bajo	Grade = alta	0,121	0,98884
Quizzes = alto ∧ Timetoassignments = bajo Timetofirst = bajo	Grade = alta	0,121	0,98884
Resources = bajo ∧ Urls = medio	Grade = media	0,091	0,98476
Resources = medio ∧ Timetofirst = alto	Grade = baja	0,091	0,98476
Resources = alto ∧ Timetoresources = bajo	Grade = alta	0,091	0,98476
Urls = bajo ∧ Timetoquizzes = alto	Grade = baja	0,091	0,98476
Quizzes = medio ∧ Timetoassignments = medio	Grade = baja	0,091	0,98476
Quizzes = medio ∧ Timetofirst = bajo	Grade = media	0,091	0,98476
Timetoquizzes = medio ∧ Timetourls = medio	Grade = baja	0,091	0,98476
Timetoresources = alto ∧ Timetourls = medio	Grade = baja	0,091	0,98476
Resources = medio ∧ Urls = medio ∧ Timetoassignments = bajo	Grade = media	0,091	0,98476
Resources = alto ∧ Urls = alto Quizzes = alto	Grade = media	0,091	0,98476
Resources = alto ∧ Quizzes = alto ∧ Timetofirst = medio	Grade = media	0,091	0,98476
Urls = medio ∧ Quizzes = medio ∧ Timetoassignments = bajo	Grade = media	0,091	0,98476
Urls = alto ∧ Quizzes = alto ∧ Timetofirst = medio	Grade = media	0,091	0,98476
Resources = bajo ∧ Timetourls = bajo	Grade = media	0,061	0,97695
Resources = bajo ∧ Timetofirst = medio	Grade = baja	0,061	0,97695
Resources = medio ∧ Quizzes = bajo	Grade = media	0,061	0,97695
Resources = medio ∧ Quizzes = alto	Grade = alta	0,061	0,97695
Resources = medio ∧ Timetoassignments = alto	Grade = baja	0,061	0,97695
Resources = alto ∧ Timetourls = alto	Grade = alta	0,061	0,97695
Resources = alto ∧ Timetofirst = bajo	Grade = alta	0,061	0,97695
Urls = alto ∧ Quizzes = medio	Grade = baja	0,061	0,97695
Urls = alto ∧ Timetoresources = bajo	Grade = alta	0,061	0,97695
Urls = alto ∧ Timetofirst = bajo	Grade = alta	0,061	0,97695
Quizzes = alto ∧ Timetoresources = alto	Grade = media	0,061	0,97695
Timetoquizzes = medio ∧ Timetoresources = bajo	Grade = alta	0,061	0,97695
Timetoquizzes = alto ∧ Timetourls = bajo	Grade = alta	0,061	0,97695
Timetoresources = alto ∧ Timetoassignments = bajo	Grade = media	0,061	0,97695
Timetourls = medio ∧ Timetoassignments = medio	Grade = baja	0,061	0,97695
Resources = bajo ∧ Urls = bajo ∧ Quizzes = medio	Grade = baja	0,061	0,97695
Resources = bajo ∧ Quizzes = bajo ∧ Timetoresources = alto	Grade = baja	0,061	0,97695
Resources = bajo ∧ Quizzes = medio ∧ Timetoquizzes = medio	Grade = baja	0,061	0,97695
Resources = bajo ∧ Quizzes = alto ∧ Timetoassignments = medio	Grade = media	0,061	0,97695
Resources = bajo ∧ Timetoquizzes = medio ∧ Timetoassignments = alto	Grade = baja	0,061	0,97695
Resources = medio ∧ Urls = medio ∧ Timetoquizzes = medio	Grade = media	0,061	0,97695
Resources = medio ∧ Urls = medio ∧ Timetoresources = bajo	Grade = media	0,061	0,97695
Resources = medio ∧ Urls = medio ∧ Timetofirst = bajo	Grade = media	0,061	0,97695
Resources = medio ∧ Urls = medio ∧ Timetofirst = medio	Grade = media	0,061	0,97695
Resources = medio ∧ Urls = alto ∧ Timetoassignments = bajo	Grade = alta	0,061	0,97695
Resources = medio ∧ Quizzes = medio ∧ Timetoresources = bajo	Grade = media	0,061	0,97695
Urls = bajo ∧ Quizzes = medio ∧ Timetoquizzes = medio	Grade = baja	0,061	0,97695
Quizzes = bajo ∧ Timetoresources = alto ∧ Timetofirst = medio	Grade = baja	0,061	0,97695
Quizzes = medio ∧ Timetoquizzes = medio ∧ Timetoresources = alto	Grade = baja	0,061	0,97695
Quizzes = medio ∧ Timetoresources = bajo ∧ Timetoassignments = bajo	Grade = media	0,061	0,97695
Quizzes = alto ∧ Timetoquizzes = medio ∧ Timetoassignments = medio	Grade = media	0,061	0,97695
Timetoquizzes = bajo ∧ Timetoresources = bajo ∧ Timetourls = bajo Timetoassignments = bajo	Grade = alta	0,061	0,97695

Tabla E.8: Reglas obtenidas con el algoritmo *Predictive Apriori* al terminar la asignatura.

Apéndice F

Publicaciones

El trabajo realizado en esta tesis doctoral ha sido publicado en las siguientes revistas:

- Moisés Riestra-González, María del Puerto Paule-Ruiz, Francisco Ortín. Massive LMS log data analysis for the early prediction of course-agnostic student performance. *Computers & Education*, volumen 163, Abril 2021, doi: 10.1016/j.compedu.2020.104108
- María del Puerto Paule-Ruiz, Moisés Riestra-González, Miguel Sánchez-Santillán, Juan Ramón Pérez-Pérez. The procrastination related indicators in e-learning platforms. *Journal of Universal Computer Science*, volumen 21, número 1, páginas 7-22, Enero 2015, 10.3217/jucs-021-01-0007

Las siguientes publicaciones, aunque relacionadas en menor medida con la presente tesis doctoral, también formaron parte del trabajo de investigación realizado en los últimos años:

- María del Puerto Paule-Ruiz, Victor Manuel Álvarez-García, Juan Ramón Pérez-Pérez, Moisés Riestra-González. Voice interactive learning: A framework and evaluation. *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*, páginas 34-39, Julio 2013, doi: 10.1145/2462476.2462489
- Victor Manuel Álvarez-García, María del Puerto Paule-Ruiz, Moisés Riestra-González, Juan Ramón Pérez-Pérez. Voice interactive classroom: best practices and design strategies. *Concurrency and Computation: Practice and Experience*, volumen 24, número 16, páginas 1963-1973, Noviembre 2012, doi: 10.1002/cpe.2814
- Victor Manuel Álvarez-García, María del Puerto Paule-Ruiz, Moisés Riestra-González, Juan Ramón Pérez-Pérez. Designing Case Studies for the Voice Interactive Classroom. *2011 Eighth International Conference on Information Technology: New Generations*, Las Vegas, EE.UU., páginas 667-672, Abril 2011, doi: 10.1109/ITNG.2011.118

Referencias

[1] Rui Li, J. T. Singh, y Jennifer Bunk. Technology Tools in Distance Education: A Review of Faculty Adoption. En *EdMedia+ Innovate Learning*, páginas 1982–1987. Association for the Advancement of Computing in Education (AACE), 2018. 1

[2] Ryann K. Ellis. Field guide to learning management systems. *ASTD learning circuits*, páginas 1–8, 2009. 1

[3] Rianne Conijn, Chris Snijders, Ad Kleingeld, y Uwe Matzat. Predicting student performance from LMS data: A comparison of 17 blended courses using Moodle LMS. *IEEE Transactions on Learning Technologies*, 10(1):17–29, October 2017. 1, 2, 11

[4] Declan Dagger, Alexander O'Connor, Seamus Lawless, Eddie Walsh, y Vincent P. Wade. Service-oriented e-learning platforms: From monolithic systems to flexible services. *IEEE internet computing*, 11(3):28–35, May 2007. 1

[5] Martín Llamas, Manuel Caeiro, Manuel Castro, Inmaculada Plaza, y Edmundo Tovar. Use of LMS functionalities in engineering education. En *2011 Frontiers in Education Conference (FIE)*, páginas S1G–1. IEEE, 2011. 1

[6] Ryan S. Baker, David Lindrum, Mary Jane Lindrum, y David Perkowski. Analyzing Early At-Risk Factors in Higher Education E-Learning Courses. *International Educational Data Mining Society*, 2015. 1

[7] Bruce W. Tuckman. Relations of academic procrastination, rationalizations, and performance in a web course with deadlines. *Psychological reports*, 96(3_suppl):1015–1021, June 2005. 2

[8] Rebeca Cerezo, Miguel Sánchez-Santillán, M. Puerto Paule-Ruiz, y J. Carlos Núñez. Students' LMS interaction patterns and their relationship with achievement: A case study in higher education. *Computers & Education*, 96:42–54, May 2016. 2, 13

[9] Nikola Kadoić y Dijna Oreški. Analysis of student behavior and success based on logs in Moodle. En *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, páginas 0654–0659. IEEE, May 2018. 2

[10] Curtis R. Henrie, Robert Bodily, Ross Larsen, y Charles R. Graham. Exploring the potential of LMS log data as a proxy measure of student engagement. *Journal of Computing in Higher Education*, 30(2):344–362, July 2018. 2

[11] Leon Gerritsen. *Predicting student performance with Neural Networks*. PhD thesis, Doctoral dissertation, Tilburg University, 2017. 2, 9

[12] Leah P. Macfadyen y Shane Dawson. Mining LMS data to develop an "early warning system" for educators: A proof of concept. *Computers & education*, 54(2):588–599, 2010. 2, 8

[13] Ya-Han Hu, Chia-Lun Lo, y Sheng-Pao Shih. Developing early warning systems to predict students' online learning performance. *Computers in Human Behavior*, 36:469–478, July 2014. 2, 8, 10, 19, 31

[14] Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction*, 6(2-3):87–129, 1996. 3

[15] Ma del Puerto Paule Ruiz, Ma Jesús Fernández Díaz, Francisco Ortín Soler, y Juan Ramón Pérez Pérez. Adaptation in current e-learning systems. *Computer Standards & Interfaces*, 30(1-2):62–70, 2008. 3

[16] Nada Dabbagh y Anastasia Kitsantas. Using web-based pedagogical tools as scaffolds for self-regulated learning. *Instructional Science*, 33(5-6):513–540, 2005. 3

[17] Enrique Garcia, Cristóbal Romero, Sebastián Ventura, y Toon Calders. Drawbacks and solutions of applying association rule mining in learning management systems. En *Proceedings of the international workshop on applying data mining in e-learning (ADML 2007), Crete, Greece*, páginas 13–22. sn, 2007. 7

[18] Dejan Ljubobratović y Maja Matetić. Using LMS Activity Logs to Predict Student Failure with Random Forest Algorithm. *The Future of Information Sciences*, página 113, 2019. 8

[19] Cristobal Romero, Pedro Espejo, Amelia Zafra, Jose Raul Romero, y Sebastian Ventura. Web usage mining for predicting final marks of students that use Moodle courses. *Computer Applications in Engineering Education*, 21(1):135–146, 2013. 9

[20] Dragan Gašević, Shane Dawson, Tim Rogers, y Danijela Gasevic. Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *The Internet and Higher Education*, 28:68–84, January 2016. 9, 16, 28, 47

[21] Javier López-Zambrano, Juan A. Lara, y Cristóbal Romero. Towards Portability of Models for Predicting Students' Final Performance in University Courses Starting from Moodle Logs. *Applied Sciences*, 10(1):354, 2020. 9, 16

[22] Cristóbal Romero, Manuel-Ignacio López, Jose-María Luna, y Sebastián Ventura. Predicting students' final performance from participation in on-line discussion forums. *Computers & Education*, 68:458–472, 2013. 10

[23] Mr Farshid Marbouti y Heidi A. Diefes-Dux. Building course-specific regression-based models to identify at-risk students. *age*, 26(1), June 2015. 10

[24] Farshid Marbouti, Heidi A. Diefes-Dux, y Krishna Madhavan. Models for early prediction of at-risk students in a course using standards-based grading. *Computers & Education*, 103:1–15, December 2016. 11

[25] John Heywood. The evolution of a criterion referenced system of grading for engineering science coursework. En *Proceedings of the 2014 IEEE Frontiers in Education Conference (FIE)*, páginas 1–6, October 2014. 11

[26] Evandro B. Costa, Baldoino Fonseca, Marcelo Almeida Santana, Fabrísia Ferreira de Araújo, y Joilson Rego. Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in Human Behavior*, 73:247–256, August 2017. 11, 28

[27] Anjeela Jokhan, Bibhya Sharma, y Shaveen Singh. Early warning system as a predictor for student performance in higher education blended courses. *Studies in Higher Education*, 44(11):1900–1911, 2019. 11

[28] David Monllaó Olivé, Du Q. Huynh, Mark Reynolds, Martin Dougiamas, y Damyon Wiese. A quest for a one-size-fits-all neural network: Early prediction of students at risk in online courses. *IEEE Transactions on Learning Technologies*, 12(2):171–183, 2019. 12

[29] Nikola Tomasevic, Nikola Gvozdenovic, y Sanja Vranes. An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers & Education*, 143:103676, 2020. 12

[30] Luis Talavera y Elena Gaudioso. Mining student data to characterize similar behavior groups in unstructured collaboration spaces. En *Workshop on artificial intelligence in CSCL. 16th European conference on artificial intelligence*, páginas 17–23, 2004. 12

[31] Jui-Long Hung y Ke Zhang. Revealing online learning behaviors and activity patterns and making predictions with data mining techniques in online teaching. *MERLOT Journal of Online Learning and Teaching*, December 2008. 13

[32] Germán Cobo, David García-Solórzano, Jose Antonio Morán, Eugenia Santamaría, Carlos Monzo, y Javier Melenchón. Using agglomerative hierarchical clustering to model learner participation profiles in online discussion forums. En *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, páginas 248–251, 2012. 13

[33] Manuel Ignacio Lopez, J. M. Luna, C. Romero, y S. Ventura. Classification via clustering for predicting final marks based on student participation in forums. *International Educational Data Mining Society*, June 2012. 13

[34] Yeonjeong Park, Ji Hyun Yu, y Il-Hyun Jo. Clustering blended learning courses by online behavior data: A case study in a Korean higher education institute. *The Internet and Higher Education*, 29:1–11, 2016. 14

[35] Danial Hooshyar, Margus Pedaste, y Yeongwook Yang. Mining Educational Data to Predict Students' Performance through Procrastination Behavior. *Entropy*, 22(1):12, December 2020. 14

[36] Jason Cole y Helen Foster. *Using Moodle: Teaching with the popular open source course management system*. O'Reilly Media, Inc., second edition edition, November 2007. 15

[37] John W Tukey et al. *Exploratory data analysis*, volume 2. Reading, Mass., 1977. 19

[38] Tarald O. Kvålseth. Cautionary Note about R^2. *The American Statistician*, 39(4):279–285, 1985. 24

[39] Ivan Tomek. Two modifications of CNN. *IEEE Transactions on Systems Man and Communications SMC-6*, 6(11):769–772, 1976. 26

[40] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, y W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, June 2002. 26

[41] Neelam Rout, Debahuti Mishra, y Manas Kumar Mallick. Handling imbalanced data: a survey. En *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*, páginas 431–443. Springer, 2018. 26

[42] Salvador Garcia, Julian Luengo, José Antonio Sáez, Victoria Lopez, y Francisco Herrera. A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):734–750, February 2012. 26

[43] Richard R. Picard y Kenneth N. Berk. Data splitting. *The American Statistician*, 44(2):140–147, 1990. 26

[44] Zuzana Reitermanová. Data Splitting. En *Proceedings of the 19th Annual Conference of Doctoral Student*, WDS, páginas 31–26, 2010. 27

[45] Ishwank Singh, A. Sai Sabitha, y Abhay Bansal. Student performance analysis using clustering algorithm. En *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*, páginas 294–299. IEEE, January 2016. 28

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, y E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 28, 67, 71

[47] Jesse Davis y Mark Goadrich. The Relationship between Precision-Recall and ROC Curves. En *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, página 233–240, New York, NY, USA, 2006. Association for Computing Machinery. 29

[48] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, USA, 2nd edition, 1979. 32

[49] Francisco Ortin, Oscar Rodriguez-Prieto, Nicolas Pascual, y Miguel Garcia. Heterogeneous tree structure classification to label Java programmers according to their expertise level. *Future Generation Computer Systems*, 105:380–394, April 2020. 33

[50] Bjoern H. Menze, B. Michael Kelm, Ralf Masuch, Uwe Himmelreich, Peter Bachert, Wolfgang Petrich, y Fred A. Hamprecht. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics*, 10(1):213, 2009. 34

[51] Stefano Nembrini, Inke R. König, y Marvin N. Wright. The revival of the Gini importance? *Bioinformatics*, 34(21):3711–3718, 2018. 34

[52] Gabriel L. Schlomer, Sheri Bauman, y Noel A. Card. Best practices for missing data management in counseling psychology. *Journal of Counseling psychology*, 57(1):1, 2010. 37

[53] Susan M. Fox-Wasylyshyn y Maher M. El-Masri. Handling missing data in self-report measures. *Research in nursing & health*, 28(6):488–495, November 2005. 37

[54] Nenad Tomašev, Miloš Radovanovič, Dunja Mladenič, y Mirjana Ivanovič. The Role of Hubness in Clustering High-Dimensional Data. En *Proceedings of the 15th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume Part I*, PAKDD'11, página 183–195, Berlin, Heidelberg, 2011. Springer-Verlag. 38

[55] Pabitra Mitra, C. A. Murthy, y Sankar K. Pal. Unsupervised feature selection using feature similarity. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):301–312, 2002. 38

[56] Ian Jolliffe. Principal component analysis. *Encyclopedia of statistics in behavioral science*, 2005. 38

[57] Zheng Zhao y Huan Liu. Spectral feature selection for supervised and unsupervised learning. En *Proceedings of the 24th international conference on Machine learning*, páginas 1151–1157. ACM, June 2007. 38

[58] Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963. 39

[59] Purnima Bholowalia y Arvind Kumar. EBK-means: A clustering technique based on elbow method and k-means in WSN. *International Journal of Computer Applications*, 105(9), 2014. 39

[60] M. A. Syakur, B. K. Khotimah, E. M. S. Rochman, y B. D. Satoto. Integration k-means clustering method and elbow method for identification of the best customer profile cluster. En *IOP Conference Series: Materials Science and Engineering*, página 012017. IOP Publishing, 2018. 39

[61] Trupti M. Kodinariya y Prashant R. Makwana. Review on determining number of Cluster in K-Means Clustering. *International Journal*, 1(6):90–95, 2013. 39

[62] Robert Tibshirani, Guenther Walther, y Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, January 2001. 39

[63] Shubhendu Trivedi, Zachary A. Pardos, y Neil T. Heffernan. Clustering students to generate an ensemble to improve standard test score predictions. En *International Conference on Artificial Intelligence in Education*, páginas 377–384. Springer, 2011. 39

[64] Hervé Abdi y Lynne J. Williams. Tukey's honestly significant difference (HSD) test. *Encyclopedia of research design*, 3:583–585, 2010. 40

[65] Andy Georges, Dries Buytaert, y Lieven Eeckhout. Statistically rigorous java performance evaluation. *ACM SIGPLAN Notices*, 42(10):57–76, 2007. 44

[66] MPuerto Paule-Ruiz, Moisés Riestra González, Miguel Sánchez Santillán, y Juan Ramón Pérez Pérez. The procrastination related indicators in e-learning platforms. *Journal of Universal Computer Science*, 2015. 47

[67] Cristóbal Romero, Sebastián Ventura, y Enrique García. Data mining in course management systems: Moodle case study and tutorial. *Computers & Education*, 51(1):368–384, 2008. 49

[68] Enrique García, Cristóbal Romero, Sebastián Ventura, y Carlos de Castro. Using rules discovery for the continuous improvement of e-learning courses. En *International Conference on Intelligent Data Engineering and Automated Learning*, páginas 887–895. Springer, 2006. 49

[69] Cristóbal Romero y Sebastián Ventura. Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6):601–618, 2010. 50

[70] Rakesh Agrawal, Tomasz Imieliński, y Arun Swami. Mining association rules between sets of items in large databases. En *Proceedings of the 1993 ACM*

SIGMOD international conference on Management of data, páginas 207–216, 1993. 50

[71] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, y Ian H. Witten. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, November 2009. 50

[72] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. En *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, páginas 487–499. Citeseer, 1994. 50

[73] Tobias Scheffer. Finding association rules that trade support optimally against confidence. En *European conference on principles of data mining and knowledge discovery*, páginas 424–435. Springer, 2001. 50

[74] Stefan Mutter, Mark Hall, y Eibe Frank. Using classification to evaluate the output of confidence-based association rule mining. En *Australasian Joint Conference on Artificial Intelligence*, páginas 538–549. Springer, 2004. 50

[75] Alexander Nussbaumer, Christina Steiner, y Dietrich Albert. Visualisation tools for supporting self-regulated learning through exploiting competence structures. En *Proceedings of the international conference on knowledge management (IKNOW 2008)*, páginas 3–5. Citeseer, 2008. 59

[76] Jakub Kuzilek, Martin Hlosta, Drahomira Herrmannova, Zdenek Zdrahal, y Annika Wolff. Analyse: analysing at-risk students at The Open University. *Learning Analytics Review*, páginas 1–16, 2015. 59

[77] Fumiya Okubo, Takayoshi Yamashita, Atsushi Shimada, y Hiroaki Ogata. A neural network approach for students' performance prediction. En *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, páginas 598–599, 2017. 59

[78] Amal Asselman, Mohamed Khaldi, y Souhaib Aammou. Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. *Interactive Learning Environments*, páginas 1–20, 2021. 59

[79] Malak Abdullah, Mahmoud Al-Ayyoub, Saif AlRawashdeh, y Farah Shatnawi. E-learningDJUST: E-learning dataset from Jordan university of science and technology toward investigating the impact of COVID-19 pandemic on education. *Neural Computing and Applications*, páginas 1–15, 2021. 59

[80] Scott M. Lundberg y Su-In Lee. A unified approach to interpreting model predictions. En *Proceedings of the 31st international conference on neural information processing systems*, páginas 4768–4777, 2017. 59

[81] Marco Tulio Ribeiro, Sameer Singh, y Carlos Guestrin. Why should I trust you? Explaining the predictions of any classifier. En *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, páginas 1135–1144, 2016. 59