

# GRADEFORESEER: Recurso docente para la predicción de notas del alumnado de informática

Maria Salamó, Inmaculada Rodríguez, Maite López-Sánchez, Anna Puig, Simone Balocco,  
Mariona Taulé, Jordi Rodríguez

Universitat de Barcelona

Gran Via de les Corts Catalanes, 585, 08007 Barcelona

{maria.salamo,inmarodriguez,maite\_lopez,annapuig,simone.balocco,mtaule}@ub.edu

## Resumen

Desde la implantación del Espacio Europeo de Educación Superior (EEES) en los diferentes grados, se ha puesto de manifiesto la necesidad de utilizar diversos mecanismos que permitan evaluar automáticamente y proporcionar una retroalimentación rápida tanto al alumnado como al profesorado sobre la evolución de los alumnos en una asignatura. En este artículo se presenta GRADEFORESEER, un recurso docente para la predicción de notas basado en técnicas de aprendizaje automático que permite evaluar la evolución del alumnado de forma automática y estimar su nota final al terminar el curso. Este recurso se ha complementado con una interfaz de usuario para el profesorado que puede ser usada en diferentes plataformas software (sistemas operativos) y en cualquier asignatura del grado de Ingeniería Informática. Además de la descripción del recurso, este artículo presenta los resultados obtenidos al aplicar el sistema de predicción en las asignaturas de Programación I (PI) y Diseño de Software (DSW) del grado de Ingeniería Informática de la Universidad de Barcelona. La capacidad predictiva se ha evaluado de forma binaria (aprueba o no) o según un criterio de rango (suspense, aprobado, notable o sobresaliente), obteniendo mejores predicciones en los resultados evaluados de forma binaria: 86.96 en DSW y 85.87 en PI.

## Abstract

The implementation of the European Higher Education Area (EHEA) has highlighted the need for alternative mechanisms to automatically evaluate students' progress in a subject, and to provide both students and teachers with quick feedback on progress. This paper presents GRADEFORESEER, a teaching resource for the prediction of grades on any Computer Science degree course. It is based on machine learning techniques, evaluates students' progress and estimates their final gra-

de at the end of the course. This resource is endowed with a user interface for teachers that can be used on different software platforms (i.e., operating systems). Additionally, this paper presents predicted results in the subjects of Software Design (DSW) and Programming I (PI) in the degree of Computer Science at the University of Barcelona. Its predictive capacity was assessed using both binary (pass or fail) and range (fail, pass, good, or excellent) criteria. Binary predictions turned out to achieve best results (86.96 % and 85.87 % in DSW and PI, respectively).

## Palabras Clave

Aprendizaje automático, Sistema de predicción de notas, Herramienta docente.

## 1. Introducción

La adaptación de los distintos grados al EEES ha significado un cambio en el modelo de enseñanza universitaria donde el estudiante ha pasado a ser el sujeto activo del aprendizaje convirtiéndose en un generador de información, ya sea de datos de sus distintas actividades docentes, como de interacciones con plataformas de *e-learning*. La utilización de plataformas digitales para la interacción en el proceso de enseñanza-aprendizaje ha abierto la posibilidad de analizar esta información tanto para la reflexión sobre el propio proceso educativo como para la predicción de la evolución del estudiante. Estos dos factores son cruciales para entender el enfoque del Análisis del Aprendizaje (*Learning Analytics*), que se define como la medida, recopilación, análisis y presentación de datos sobre el alumnado (y sus contextos), y que tiene la finalidad de comprender y optimizar el aprendizaje [14].

El Análisis del Aprendizaje abarca diferentes dimensiones que caracterizan el marco de actuación, como pueden ser los usuarios a los que va dirigido o los mé-

todos utilizados [3]. En este artículo se presenta GRADEFORSEER, un nuevo recurso docente de análisis de los resultados del aprendizaje para estimar la evolución del alumnado en una asignatura, facilitando al profesorado una predicción sobre el éxito o fracaso que pueda tener cada estudiante en una asignatura. A partir de los primeros resultados de pruebas, actividades, ejercicios o prácticas de cada estudiante en una asignatura, GRADEFORSEER predice automáticamente el resultado final más probable del estudiante. El profesorado puede rectificar y personalizar la curva de aprendizaje del alumnado en etapas muy preliminares, reforzando los puntos débiles con enlaces de interés, referencias bibliográficas, apuntes, tutorías personalizadas, ejercicios resueltos, etc. En este sentido, nuestra propuesta no substituye al profesorado sino que detecta qué alumnado no sigue de forma satisfactoria los objetivos de aprendizaje para que el profesor pueda incidir en su planificación docente, a diferencia de los Sistemas Tutores Inteligentes [2].

Este primer estudio se ha centrado en los primeros cursos del grado de Ingeniería Informática, en los que el abandono y el fracaso es elevado. Actualmente, el alumnado proviene de estudios de Bachillerato y de módulos de Grado Superior, con unas dinámicas de trabajo y retroalimentación de sus actividades muy diferentes a las implantadas en la universidad. En estos estudios previos, el ratio de alumnos por aula es muy reducido, los alumnos tienen numerosas pruebas, una o varias entregas semanales de pequeños ejercicios, y una planificación muy guiada de los trabajos que deben realizar. Al llegar a la universidad, el ratio de alumnos por aula es superior, las pruebas están más dispersas en el tiempo, las entregas de ejercicios prácticos son a medio plazo y se promueve que cada estudiante planifique su trabajo de forma autónoma. Este cambio de dinámica es un factor esencial que influye en la pérdida de motivación y el consecuente abandono de estudiantes en las asignaturas de los primeros cursos. GRADEFORSEER se ha analizado en las asignaturas de Diseño de Software y Programación I del grado de Ingeniería Informática de la Universidad de Barcelona. Cabe destacar que el sistema es fácilmente generalizable a otras asignaturas del grado, siempre que el plan docente de éstas asignaturas contemple la realización de diversas pruebas de evaluación (prácticas, parciales, presentaciones orales, etc.) a lo largo del curso.

Para facilitar la utilización del sistema y la visualización de la predicción, el sistema se ha complementado con una aplicación multiplataforma dirigida al profesorado que puede ser utilizada en cualquier contexto (diferentes asignaturas).

La organización del presente artículo es la siguiente. En el apartado 2, se contextualiza el recurso docente. En el apartado 3, se describe la propuesta de evaluación

de la evolución del alumnado, en concreto, el sistema de predicción y la interfaz de usuario. En el apartado 4, se presentan los resultados obtenidos en las dos asignaturas del grado de Ingeniería Informática. Finalmente, en el apartado 5 se presentan las conclusiones y las futuras líneas de investigación.

## 2. Trabajo relacionado

Dentro del ámbito docente, el abanico de posibilidades que se abre a la hora de introducir herramientas software para hacer un seguimiento del alumnado es realmente amplio y diverso [10, 2, 15]. En este apartado no se pretende cubrir de forma exhaustiva dicha variedad, sino mencionar algunos de los trabajos que son relevantes para contextualizar y diferenciar la aportación del recurso docente que aquí se presenta.

Los sistemas más populares dedicados al seguimiento del alumnado son los Sistemas Tutores, que al incluir técnicas del área de Inteligencia Artificial son denominados Sistemas Tutores Inteligentes [2, 11]. En general, estos sistemas pretenden substituir el tutor (entendido como profesor, alguien que guía al alumnado en el aprendizaje de una materia), de manera que es el propio sistema el que marca la forma en la que el alumnado adquiere nuevas competencias y conocimientos y que evalúa si éste alcanza los objetivos de aprendizaje marcados. Sin embargo, en el caso de nuestro recurso docente GRADEFORSEER, es importante destacar que su objetivo es ayudar al profesorado en el seguimiento de la evolución del alumnado proporcionándole indicadores de su rendimiento académico, y alertándole cuando se prevee que dicho rendimiento no es el esperado, para que el profesorado pueda tomar medidas preventivas. Por tanto, este recurso no substituye en ningún caso al profesorado, ya que requiere que éste siga llevando a cabo sus tareas docentes (tanto de acompañamiento en la adquisición de competencias y conocimiento como de evaluación).

Siguiendo esta misma línea de discurso, es importante diferenciar el recurso docente propuesto en este artículo del aprendizaje electrónico (*e-learning*) [15], un área que fusiona las TIC (Tecnologías de la Información y Comunicación) y la educación para permitir, en primera instancia, el aprendizaje autónomo (autoaprendizaje) a distancia. Como no podía ser de otro modo, los Sistemas Tutores constituyen componentes fundamentales en los sistemas de aprendizaje electrónico [9], y de la misma forma que hablamos de Sistemas Tutores Inteligentes, es posible aplicar técnicas de Inteligencia Artificial (como por ejemplo, la personalización) en los sistemas de aprendizaje electrónico [6].

Sin embargo, si nos centramos en el seguimiento del alumnado sin incidir en las tareas típicas del profesor (enseñanza y evaluación), encontramos que el concepto

de tutor se entiende también como de orientador individualizado, alguien que pueda detectar problemas y disparar las alarmas necesarias para la reorientación del alumnado [1]. Ésta es la aproximación que toma nuestro recurso docente y que se localiza dentro del área del Análisis del Aprendizaje [14]. Éste enfoque se puede abordar realizando el seguimiento dentro de una asignatura concreta (tal y como es en nuestro caso<sup>1</sup>) o, de forma transversal a los estudios que se estén realizando. Esta última aproximación se está empezando a implantar en algunas universidades [5], aunque su implantación es aún discreta, ya que el *Education Advisory Board* de Estados Unidos calcula que únicamente unas 125 instituciones universitarias del total de las 4000 (aprox.) existentes aplican técnicas de análisis del aprendizaje para seguir la evolución del alumnado a lo largo de los estudios. No obstante, ya se han aportado resultados positivos respecto a la reducción de alumnado que abandona los estudios [7].

### 3. GRADEFORESEER

En esta sección vamos a detallar cómo se ha abordado el recurso docente desde la perspectiva de un problema de predicción y se ha resuelto mediante diferentes técnicas de aprendizaje automático definidas en un módulo de predicción independiente. Además, se introduce la interfaz de usuario GRADEFORESEER-I, que permite al profesorado solicitar y obtener una predicción de notas de los alumnos de sus asignaturas.

#### 3.1. Sistema de predicción

Tal y como se muestra en la Figura 1, la idea principal del módulo de predicción es utilizar los datos históricos de una asignatura que contengan los resultados del alumnado a lo largo de uno o más cursos académicos y, mediante diferentes algoritmos de clasificación<sup>2</sup>, predecir la nota final que obtendrá el alumnado del curso académico actual.

El objetivo es por tanto predecir la nota de los alumnos en un instante del curso académico en el que ya se han realizado diversas <sup>3</sup> pruebas de cualquier tipo que evalúen competencias básicas o transversales, ya sean prácticas, parciales o presentaciones en clase, pero que tengan asignada una calificación. De esta forma, por ejemplo a mitad de curso, el profesorado puede analizar la evolución de sus alumnos y así realizar las acciones

<sup>1</sup>El módulo de aprendizaje propuesto ha sido desarrollado dentro de un proyecto de innovación docente [12].

<sup>2</sup>Un algoritmo de clasificación tiene como objetivo asignar un elemento entrante no etiquetado en una categoría concreta conocida.

<sup>3</sup>Aunque el módulo de predicción funcionará mejor con un mayor número de notas, las evaluaciones realizadas demuestran que un 50 % del total es suficiente para obtener resultados aceptables.

necesarias para reforzar los puntos más débiles detectados en aquellos alumnos susceptibles de suspender la asignatura. Asimismo, la predicción de notas puede ser una herramienta para motivar a alumnos que, con las notas que tienen actualmente, ven difícil aprobar. De esta forma, se aborda la frustración que sufren los alumnos que obtienen calificaciones bajas en las primeras pruebas del curso.

Para el desarrollo de GRADEFORESEER se ha utilizado Java, debido a que tiene como ventaja que es multiplataforma y se espera que el recurso docente sea utilizado por diferentes usuarios en diferentes plataformas software. Asimismo, las librerías de aprendizaje automático que incorporan los diferentes clasificadores también están implementadas en Java, lo que nos ha permitido el desarrollo de un módulo homogéneo.

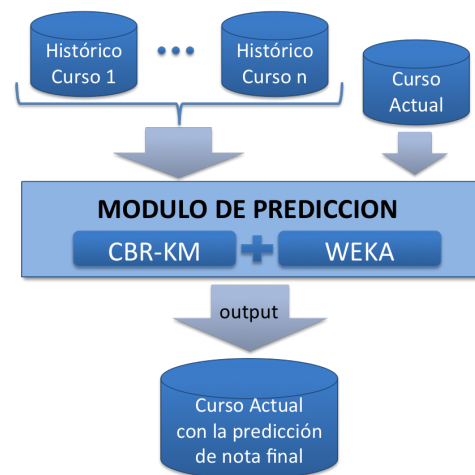


Figura 1: Diagrama conceptual del sistema predictivo.

En concreto, tal y como se puede observar en el diagrama conceptual de la Figura 1, el módulo de predicción realiza tres pasos: (1) recoge los datos de entrada; (2) entrena un conjunto de clasificadores con los datos recibidos; y (3) genera las predicciones y las vuelca en un fichero de salida.

En el primer paso, el módulo de predicción recoge como datos de entrada tanto los datos históricos completos de una asignatura, los cuales serán el conjunto de entrenamiento de los diferentes clasificadores, como los datos incompletos de los alumnos actuales, que corresponderán con el conjunto a predecir por cada clasificador. El histórico de cada curso académico (de cada asignatura) contiene un conjunto de alumnos y para cada uno de ellos, el conjunto de notas obtenidas en los ejercicios, prácticas, pruebas o exámenes, así como la nota final obtenida en dicha asignatura al finalizar el curso.

En el segundo paso, a partir de estos datos (el histórico y los datos a predecir) el módulo de predicción

ejecuta diversos algoritmos de clasificación. Éstos provienen de dos librerías de aprendizaje automático que se han unificado en el módulo: CBR-KM[8] y WEKA[4], cada una de ellas con un conjunto de algoritmos de clasificación.

- CBR-KM (*Case-Based Reasoning-Knowledge Management*), es una librería Java de aprendizaje automático desarrollada en la Universidad de Barcelona que incorpora un conjunto de algoritmos de clasificación de Razonamiento Basado en Casos y algoritmos específicos para el mantenimiento de la memoria de casos, entre ellos el *CBR Adaptativo (ACBR)* [13].
- WEKA (*Waikato Environment for Knowledge Analysis*), es una librería Java de aprendizaje automático desarrollada por la Universidad de Waikato que es de software libre bajo licencia GNU. Esta librería contiene algoritmos para el preprocesado de los datos, algoritmos de clasificación y de clusterización, diferentes métodos de evaluación, además de una interfaz para visualizar datos y comparar algoritmos. En el desarrollo de nuestro módulo de predicción se han usado tanto los algoritmos para el preprocesado de los datos, como los algoritmos de clasificación disponibles en la librería. Esta librería incorpora un total de 87 algoritmos de clasificación provenientes de WEKA como por ejemplo algoritmos de redes bayesianas, redes neuronales, árboles de decisión, algoritmos perezosos, etc.

En el tercer paso, el módulo de predicción extrae un fichero de datos con la nota final estimada para cada alumno del curso actual. Actualmente esta nota se predice haciendo una validación cruzada (ver más detalles en sección 4.2.1) del histórico (conjunto de entrenamiento) y, a partir del mejor clasificador obtenido en ésta validación cruzada, se procede a predecir la nota final de los alumnos del curso actual.

Finalmente, cabe mencionar que la idea de usar un sistema clasificador para predecir la evolución de un alumno ya se planteó en un trabajo anterior [16]. En concreto se analizó como un clasificador de Razonamiento Basado en Casos predecía la evolución de los alumnos en la asignatura de Programación I. A diferencia del trabajo anterior, en el que se analizó un único clasificador y se desarrolló para una asignatura en concreto, este artículo presenta un recurso docente desarrollado en el contexto de un proyecto de innovación docente [12] que se adapta a cualquier asignatura, que engloba un módulo con un conjunto de clasificadores provenientes de dos librerías para realizar la predicción, y que incluye, además, una interfaz que facilita el uso al profesorado<sup>4</sup>.

<sup>4</sup>El código del recurso docente se encuentra disponible en

### 3.2. Interfaz GRADEFORSEER-I

Esta sección se presenta GRADEFORSEER-I, una aplicación para predecir la nota final del alumnado de una asignatura. Se trata de un primer prototipo que tiene como usuarios principales a profesores de informática, pero que a medida que se abra a toda la comunidad educativa ha de transformar su interfaz para adaptarse a un perfil de usuario menos técnico.

GRADEFORSEER-I es una aplicación de escritorio multiplataforma, desarrollada en Java, que utiliza Java Swing para la GUI (*Graphical User Interface*). Durante la instalación de la aplicación se permite seleccionar el idioma de la interfaz, y se instala automáticamente la librería de clasificadores de WEKA y CBR-KM, necesaria para su correcta ejecución.

Para que la aplicación pueda realizar la predicción de notas de los alumnos de una asignatura, ésta necesita primero que el profesor proporcione i) las notas de las diferentes actividades evaluativas de la asignatura en cursos anteriores y ii) las notas que tiene de dicha asignatura en el curso actual. El objetivo es averiguar, con los datos que se disponen hasta el momento del curso actual (por ejemplo, las 3 primeras prácticas y la primera prueba parcial teórica), cual podría ser la nota final de los alumnos.

Estos datos se deben proporcionar en los formatos CSV o XLS, en un fichero con nombre <NombreAsignatura>\_Curso\_<AñoInicial>\_<AñoFinal>.xls. Por ejemplo, el fichero de la asignatura de Diseño de Software del curso 2013-2014 se llamará "DSW\_Curso\_2013\_2014.xls". Tal y como se ha mencionado, estos ficheros deben contener para cada alumno la nota obtenida en cada una de las actividades evaluativas (prácticas, pruebas parciales, finales).

Una vez instalada la aplicación, y suponiendo que el profesor tiene preparados los ficheros CSV o XLS que se han mencionado previamente, éste puede comenzar a interactuar con la misma realizando los siguientes pasos:

1. Cargar los datos históricos: uno o más ficheros con las notas de cursos académicos anteriores.
2. Cargar los datos actuales: un fichero con las notas del curso actual.
3. Solicitar la predicción. El profesor puede seleccionar la opción *Predicción rápida* o bien *Predicción exhaustiva* (más lenta).
4. Visualizar el resultado de la predicción.

En la Figura 2, se muestra cómo la interfaz proporciona al profesor los elementos de interacción necesarios para realizar los pasos anteriores.

En primer lugar, la parte dedicada a los "Ficheros cargados actualmente" en la figura muestra dos pestañas que permiten cargar o bien las notas de cursos

<http://wai.maia.ub.es/projects.htm> (Educational innovation)

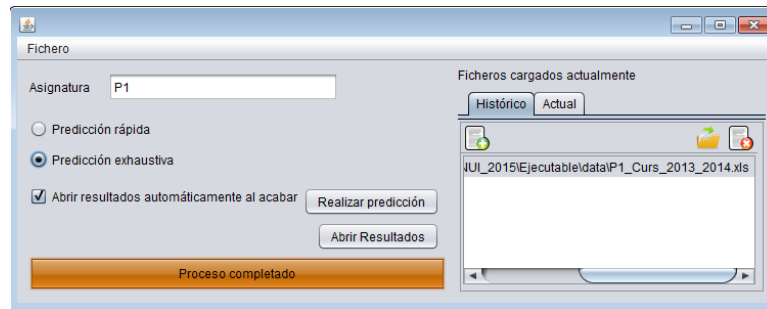


Figura 2: Interfaz de la aplicación.

previos (pestaña Histórico) o bien las notas de que se dispone hasta el momento (pestaña Actual). El profesor utiliza el icono visual que muestra el signo +, o la correspondiente opción de menú, para añadir un fichero de notas. El profesor también puede eliminar ficheros ya incluidos mediante el icono visual que muestra una x en color rojo.

Una vez cargados los ficheros de notas, el profesor está en disposición de solicitar la predicción. Para ello, tiene dos opciones. Por un lado, la predicción rápida, que utiliza un único sistema de predicción, la librería CBR-KM presentada en la sección 3.1. Por otro lado, la predicción exhaustiva, que realiza pruebas de validación cruzada con el conjunto de datos de entrenamiento (datos de cursos anteriores) con toda la batería de clasificadores de WEKA y todos los posibles parámetros del algoritmo clásico de CBR y, se queda con el clasificador que ha dado un porcentaje de aciertos más alto para realizar la predicción del curso actual. Esta segunda opción es, por tanto, más lenta pero podría dar resultados más precisos.

Mientras que el sistema está realizando la predicción, hay una barra de progreso que proporciona al profesorado un feedback visual sobre el grado de compleción de la operación. En la Figura 2 se muestra la barra completa dado que ha finalizado el proceso de predicción.

Una vez terminada la ejecución de los algoritmos de predicción, hay una opción para que se abra automáticamente el fichero que contiene la predicción de notas (véase la Figura 3). Este fichero contiene una nueva columna con la nota prevista: Suspenso, Aprobado, Notable o Sobresaliente. Finalmente, el profesor puede guardar los datos de predicción generados por la aplicación.

## 4. Evaluación en dos asignaturas

La evaluación del sistema de predicción de notas se ha realizado con dos asignaturas del grado de Informática de la Universidad de Barcelona. Este grado se desarrolla a lo largo de 4 cursos académicos y las asignaturas son Programación I (formación básica) y

	A	B	C	D	E	G
1	ID_user	E1	E2	E3	Parcial1	NOTA PREVISTA
2	1	10	1	1	3,5	Aprobado
3	2	0	0	0	0	Suspenso
4	3	0	6	0	5	Aprobado
5	4	4	0	0	0	Suspenso
6	5	4	8,5	7,5	7	Aprobado
7	6	10	9	9	8	Excelente
8	7	0	0	0	3,75	Suspenso
9	8	9	0	0	2,75	Aprobado
10	9	10	9,8	9,75	6,55	Excelente
11	10	5	9,5	7,5	2,35	Aprobado
12	11	0	3	0	3,75	Suspenso
13	12	0	0	0	0	Suspenso
14	13	5	7	4,5	3,25	Suspenso
15	14	0	0	0	0	Suspenso
16	15	7,8	9,5	0	4	Suspenso
17	16	7	5	5,5	5	Suspenso
18	17	2	0	0	0	Suspenso
19	18	6,6	8	9	5,5	Notable

Figura 3: Fichero con una nueva columna que muestra el resultado de la predicción.

Diseño de Software (obligatoria), asignaturas de primer y segundo curso, respectivamente.

### 4.1. Descripción de las asignaturas

Programación I tiene como objetivo iniciar al estudiante en el mundo de la programación para resolver problemas científicos y técnicos de forma metódica y sistemática. La asignatura se imparte en clases teóricas (2h semanales) y clases prácticas (2h semanales). Las clases teóricas se dedican a la exposición del temario y a realizar ejercicios en pizarra. Las clases prácticas se realizan en aulas de informática donde los alumnos realizan 4 trabajos prácticos (*P1*, *P2*, *P3*, *P4*) a lo largo del curso.

Las actividades de evaluación que se realizan en esta asignatura constan de una nota de teoría y una nota de prácticas. La nota de teoría se calcula con el resultado obtenido en dos pruebas parciales teóricas a lo largo del curso (*Parcial1*, *Parcial2*) y la nota de prácticas se calcula con los resultados obtenidos en cada una de las prácticas entregadas. Por tanto, el histórico de notas que se puede proporcionar al sistema de predicción consta de: 4 prácticas, 2 notas de pruebas teóricas y la nota final.

La asignatura de Diseño de Software tiene como

objetivo mejorar las capacidades de programación e introducir al alumno en los conceptos básicos del diseño de software. Ésta asignatura viene a continuación de las asignaturas de formación básica (como por ejemplo Programación I) y, en el plan de estudios, es la primera asignatura dedicada a la Ingeniería del Software. Así pues, la asignatura de Diseño de Software representa un reto para los alumnos dado que están a caballo entre afianzar sus conocimientos en programación y empezar a desarrollar problemas de mayor envergadura.

La asignatura de Diseño de Software se imparte en clases teóricas (2h semanales) y clases prácticas (2h semanales). Para las prácticas se sigue el aprendizaje basado en problemas, donde se proporciona a los alumnos un problema de tamaño medio que se va resolviendo siguiendo principalmente las etapas de desarrollo del software del proceso unificado. En esta asignatura se realiza una evaluación continuada de los alumnos, la cual consta de tres entregas de pequeños ejercicios prácticos ( $E1$ ,  $E2$ ,  $E3$ ), tres entregas de prácticas ( $P1$ ,  $P2$ ,  $P3$ ) y dos exámenes (uno *Parcial* a mitad del curso y otro *Final* al acabar el curso). Por lo tanto, el histórico de notas de la asignatura consta de: 3 notas de ejercicios, 3 prácticas, 2 notas de pruebas teóricas y la nota final.

## 4.2. Resultados

Para identificar los clasificadores óptimos en cada asignatura y cuantificar el rendimiento de los mismos se han realizado dos experimentos numéricos: el primero evalúa los diferentes clasificadores con validación cruzada de datos de un curso académico y el segundo evalúa la capacidad predictiva de los mejores clasificadores en cada una de las asignaturas.

En concreto, en cada uno de los experimentos se han entrenado los clasificadores usando los resultados de las asignaturas del penúltimo año, y se han evaluado los resultados sobre las notas del último año académico. Para cada experimento se han usado las dos bases de datos disponibles —Programación I (PI) y Diseño de Software (DSW)— y se han ejecutado un total de 87 clasificadores disponibles en las dos librerías CBR-KM y WEKA.

### 4.2.1. Evaluación de los clasificadores

El objetivo del primer experimento es analizar qué clasificador es el que mejor predice en cada asignatura.

Para evaluar los resultados con un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y testeo, se ha utilizado la validación cruzada de 10 iteraciones (*10 fold cross-validation*). Esta técnica consiste en utilizar una décima parte del conjunto de datos de entrada como datos de

prueba y el resto (el 90 %) como datos de entrenamiento. El proceso de validación cruzada es repetido durante 10 iteraciones, con cada uno de los posibles subconjuntos de datos de testeo. En cada iteración se calcula el valor de exactitud que corresponde al número de aciertos sobre el número total de alumnos. Finalmente se realiza la media aritmética de los valores de exactitud obtenidos en cada iteración para obtener un único resultado.

Mejor clasificador	DSW		PI	
	Rango	Binario	Rango	Binario
#1	MP	MPCO	CBR	CBR
Exactitud	69.88	91.57	81.11	92.22
#2	MPCS	MCCU	RFC	RBF
Exactitud	69.88	91.57	80	92.22
#3	OCC	RF	NBT	FNN
Exactitud	69.88	91.57	80	92.22
#4	RF	PART	SMO	KNN
Exactitud	68.67	91.57	78.89	92.22
#5	SMO	C45DT	FNN	MCCU
Exactitud	67.47	91.57	78.89	91.11

Cuadro 1: Resultados de los cinco mejores clasificadores evaluados sobre cada asignatura (DSW, PI)

En el cuadro 1, para cada asignatura considerada, se muestran los cinco mejores resultados de clasificación, ordenados de forma descendente desde el mejor resultado hasta el peor. Además, se ha evaluado separadamente la capacidad del clasificador para predecir la nota final del alumno siguiendo dos criterios: por *rango* y *binario*. En el primer caso (*rango*), se han definido rangos de valores numéricos de notas  $\{[1, 5]; [5, 7[; [7, 9[; [9, 10]\}$  correspondientes a las calificaciones de  $\{Suspense, Aprobado, Notable, Sobresaliente\}$ , respectivamente. En el segundo (criterio *binario*), sólo se ha considerado si el estudiante aprueba o no la asignatura, es decir si su nota final es superior o inferior a 5.

En el cuadro 2 se han indicado los acrónimos de los clasificadores que han obtenido mejores resultados.

Se puede observar que los resultados de predicción son particularmente buenos en el cuadro 1. Cuando el sistema de predicción evalúa los resultados de forma binaria se obtiene una exactitud superior al 90 % en ambas asignaturas. Por otro lado, cuando se evalúa por rango de notas, los resultados obtenidos son de un 69.88 % en el caso de DSW y del 81.11 % en el caso de PI.

Nombre clasificador	Acrónimo
Multilayer Perceptron Classifier Optimized	MPCO
Multilayer Perceptron	MP
Multilayer Perceptron Context Sensitive	MPCS
Randomizable Filtered Classifier KNN	RFC
Naive Bayes	NB
Naive Bayes Decision Tree	NBT
K-nearest-neighbor	KNN
CBR Classifier	CBR
Multinomial Logistic Regression Model	MLOG
Rotation Forest	RF
Functional Trees	FT
Averaged N-Dependence Estimator 2	A2DE
First Nearest-Neighbour	FNN
Decorate	DEC
Bayes Network	BN
Naive Bayes Simple	NBS
Simple Logistic	SL
Dagging – Decision Stump	DDS
Bagging – REPTree	BRT
Filtered KNN	FKNN
Multi Class Classifier Updateable	MCCU
RBF Classifier	RBF
Ordinal Class Classifier	OCC
Partial C4.5 Decision Tree	PART
Sequential Minimal Optimization	SMO
C4.5 Decision Tree	C45DT

Cuadro 2: Acrónimo de los clasificadores utilizados en el cuadro 1

#### 4.2.2. Evaluación de la capacidad predictiva del clasificador

El interés del sistema predictivo es poder analizar los datos disponibles a lo largo de la asignatura, y predecir la nota final del estudiante antes de disponer de todas las notas en un curso académico. Por este motivo, el objetivo del segundo experimento es analizar la capacidad predictiva del sistema en el caso que únicamente una parte de los resultados estén disponibles.

Para cada asignatura considerada, el conjunto de testeo ha sido creado a partir de las notas del último año académico, omitiendo las últimas columnas de notas del curso. De este modo se han eliminado el 33 % de los datos disponibles. A partir de estos datos el sistema produce una predicción de nota final de la asignatura, que se compara con las notas finales disponibles antes de la eliminación de datos.

En el cuadro 3 también se presentan los cinco mejores resultados de clasificación obtenidos en las dos asignaturas, ordenados de forma descendente desde el mejor resultado hasta el peor. Como en el caso anterior, se ha evaluado por separado la capacidad del clasifica-

dor para predecir la nota final del alumnado siguiendo dos criterios: por *rango* y *binario*.

Mejor clasificador	DSW		PI	
	Rango	Binario	Rango	Binario
#1	FT	MPCO	BN	BN
Exactitud	66.67	86.96	60.87	85.87
#2	MP	RF	NB	A2DE
Exactitud	66.67	86.96	60.87	85.87
#3	MPCS	SL	NBS	DDS
Exactitud	66.67	85.51	60.87	85.87
#4	MLOG	DEC	FKNN	BRT
Exactitud	65.22	85.51	60.87	84.78
#5	MPCO	NBT	CBR	DEC
Exactitud	63.77	85.51	60.78	84.78

Cuadro 3: Resultados de los cinco mejores clasificadores evaluados sobre cada asignatura con datos parciales

En este experimento, los resultados de predicción son aceptables, aunque la exactitud obtenida en la predicción es ligeramente inferior a la propia del clasificador. Cuando el sistema de predicción evalúa los resultados de forma binaria, se obtiene un 86.96 % de exactitud en caso de la asignatura DSW y 85.87 % para la asignatura PI. Por otro lado, cuando se evalúa por rango de notas, los resultados mejores llegan a un 66.67 % en el caso de DSW, y del 60.87 % en el caso de PI.

## 5. Conclusiones

En este artículo se ha presentado GradeForeseer, un recurso docente que permite evaluar de manera automática la evolución del alumnado en el aprendizaje de una asignatura y realizar una estimación de la nota final que obtendría al finalizar el curso.

El recurso docente se ofrece al profesorado en forma de una herramienta multiplataforma denominada GRADEFORSEER. Ésta ha sido desarrollada en Java e instala automáticamente las librerías CBR-KM y WEKA. GRADEFORSEER proporciona una interfaz simple y fácil de usar, y permite detectar en etapas iniciales-medias del curso la nota final de los alumnos. Por un lado, se pueden reforzar los conocimientos y plantear nuevas actividades a aquellos alumnos que tienen como nota prevista un suspenso, y por otro lado se puede motivar a aquellos alumnos que a pesar de tener notas bajas en las primeras actividades del curso, tienen como nota prevista un aprobado.

El núcleo de dicho recurso es un módulo de predicción basado en técnicas de aprendizaje automático que utiliza 87 algoritmos de clasificación procedentes de las librerías CBR-KM y WEKA. Los clasificadores usan

como datos de entrenamiento los ficheros con los datos completos de los cursos académicos anteriores de la misma asignatura y como datos a predecir los ficheros con los datos incompletos del curso académico actual.

En la evaluación del sistema de predicción con dos asignaturas del grado de informática, Diseño de Software (DSW) y Programación I (PI), se ha obtenido las mejores predicciones en los resultados evaluados de forma binaria (aprueba o no la asignatura), siendo éstos de un 91.57 % y un 92.22 % de acierto en DSW y PI, respectivamente. Además, la evaluación de la capacidad predictiva del sistema de predicción con notas parciales ha mostrado un alto porcentaje de aciertos (un 86.96 % en DSW y un 85.87 % en PI). Nótese que en los resultados sólo se han usado el 50 % de las pruebas del curso académico actual para predecir si un alumno aprueba o no la asignatura. Asimismo, los resultados evaluados según el rango son también aceptables, ya que se llega al 69.88 % en DSW y al 81.11 % en PI en el caso de una evaluación completa del curso académico y al 66.67 en DSW y 60.87 en PI para la evaluación con el 50 % de los datos del curso actual.

Como trabajo futuro se plantea la integración en plataformas de aprendizaje (Moodle), analizar el uso de políticas alternativas en la generación de la predicción de notas, y detectar qué pruebas de evaluación a lo largo del curso inciden más en la nota final.

## Agradecimientos

Este trabajo ha sido posible gracias a los proyectos *Módulo pedagógico en un Sistema Tutor Inteligente para predecir la evolución del alumnado* (2014PID-UB030), SGR-623-2014, TIN2012-38603-C02, TIN2011-24220 y TIN2012-38876-C02-02.

## Referencias

- [1] P. Fernández Sánchez. *La función tutorial*. Madrid: Castalia-M. E. C, 1991.
- [2] A. Graesser, P. Chipman, B. Haynes, and A. Olney.
- [3] A. Guerten. Turning learning into numbers a learning analytics framework graphic.
- [4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.
- [5] L. Igual and et al. *Proyecto de Innovación Docente: Sistema Inteligente de soporte al tutor de estudios*. Universitat de Barcelona, <http://mid.ub.edu/webpamid/content/sistema-intel·ligent-de-suport-al-tutor-d'estudis>, 2014.
- [6] M. K. Khribi, M. Jemni, and O. Nasraoui. Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. In *ICALT*, pages 241–245. IEEE, 2008.
- [7] J. Marcus. *Here's the New Way Colleges Are Predicting Student Grades*: <http://time.com/3621228/college-data-tracking-graduation-rates/>. Time, Dec 2014.
- [8] I. Martí Diéguez. CBR-KM: Plataforma de Raonament Basat en Casos per a la gestió del coneixement. Technical report, <http://hdl.handle.net/2445/62504>, MAiA dept, Universitat de Barcelona, 2014.
- [9] C. M. Pagano. Los tutores en la educación a distancia. un aporte teórico: <http://www.uoc.edu/rusc/4/2/dt/esp/pagano.pdf>. *Universitat Oberta de Catalunya*.
- [10] J. Reynol. The relationship between frequency of facebook use, participation in facebook activities, and student engagement. *Computers & Education*, 58(1):162 – 171, 2012.
- [11] M.M. Rodrigo, R.S. de Baker, J. Agapito, J. Nabos, M.C. Repalam, S. Reyes, and M.O. Pedro. The effects of an interactive software agent on student affective dynamics while using an intelligent tutoring system. *IEEE Transactions on Affective Computing*, 3(2):224–236, 2012.
- [12] M. Salamó and et al. *Proyecto de Innovación Docente: Módulo Pedagógico de un sistema Tutor Inteligente para predecir la evolución del alumnado*. Universitat de Barcelona, [http://mid.ub.edu/webpamid/content/mòdul-pedagògic-en-un-sistema-tutor-intel·ligent-pedir-l'evolució-de-l'alumnat](http://mid.ub.edu/webpamid/content/mòdul-pedagògic-en-un-sistema-tutor-intel·ligent-predir-l'evolució-de-l'alumnat), 2014.
- [13] M. Salamó and M. López-Sánchez. Adaptive case-based reasoning using retention and forgetting strategies. *Know.-Based Syst.*, 24(2):230–247, mar 2011.
- [14] G. Siemens. Learning analytics: The emergence of a discipline. *American Behavioral Scientist*, 57(10):1380–1400, 2013.
- [15] P. Sun, R. J. Tsai, G. Finger, Y. Chen, and D. Yeh. What drives a successful e-learning? an empirical investigation of the critical factors influencing learner satisfaction. *Computers & Education*, 50(4):1183 – 1202, 2008.
- [16] D. Vernet, M. Salamó, C. Vallespí, J. Camps, E. Golobardes, and J. Bacardit. ¿Cómo predecir la evolución del alumno? . In *VII Jornadas Enseñanzas Universitarias de Informática*, pages 329–334, España.