Actas de las Jenui, vol. 4. 2019.
Páginas: 311-318

Sistema para la detección temprana de anomalías en la evaluación usando técnicas de aprendizaje automático

Juan Ramón Rico-Juan Antonio-Javier Gallego
Francisco J. Castellanos Jorge Calvo-Zaragoza
Dpto. Lenguajes y Sistemas Informáticos
Universidad de Alicante
03690 Alicante
juanramonrico@ua.es jgallego@dlsi.ua.es
fcastellanos@dlsi.ua.es jcalvo@dlsi.ua.es

Resumen

Uno de procesos más importantes en casi todos los modelos de enseñanza universitaria es la evaluación. Los criterios que se establecen en una asignatura orientan la forma en la que se obtiene la calificación final del alumno. Por este motivo es importante realizar un seguimiento continuado del aprendizaje del estudiante y de sus calificaciones, permitiendo de este modo la detección de anomalías para proceder con una intervención inmediata que permita corregir la situación. Normalmente, en los primeros cursos universitarios el número de alumnos es elevado, lo que redunda en el detrimento del seguimiento que se le puede realizar a los estudiantes por parte del profesor. En este trabajo se propone un sistema para predecir la calificación de un estudiante en una determinada actividad, de forma que se notifique al profesor cuando la calificación real se aleje suficientemente del valor predicho. Para esto se ha realizado un estudio de 24 algoritmos de inteligencia artificial, seleccionando finalmente los más adecuados para el caso de estudio realizado. Los resultados experimentales muestran la utilidad del método propuesto y cómo los algoritmos basados en máquinas de vectores soporte o los de aumentado de gradiente extremo son los que mejores resultados obtienen.

Abstract

One of the most important processes in almost all university education models is evaluation. The criteria established in a subject guide how the student's final grade is obtained. Therefore, it is important to continuously monitor the student's learning process and grades, thus allowing the detection of anomalies to proceed with an immediate intervention to correct the situation. Typically, the first university courses have a high number of students, which is detrimental to the tracking that can be done by the teacher. In this paper, we propose an approach to predict the next grade of a student in a certain activity, so that the teacher is notified in case the actual grade is different enough from the predicted one. To this end, an experimental study of 24 artificial intelligence algorithms, selecting the most suitable ones for our case of study. The experimental results show the goodness of the proposed approach, and that the algorithms based on support vector machines or those of extreme gradient augmentation are the ones that best fit the considered data.

Palabras clave

Aprendizaje automático, detección de anomalías, predicción de notas, evaluación.

1. Introducción

Cada vez es más habitual encontrar publicaciones que exploran la posibilidad de aplicar técnicas de aprendizaje automático —área de la inteligencia artificial que estudia cómo pueden aprender los ordenadores a partir de datos— para prever problemas e intentar corregirlos antes de que sucedan [2]. Por ejemplo, predecir el fracaso académico de los estudiantes en los cursos de programación introductoria [6] o predecir si un estudiante finalizará satisfactoriamente o no su título universitario [8].

Si consideramos los estudios universitarios actuales no cabe duda que uno de los procesos más importantes en los modelos de enseñanza es la evaluación. Los criterios establecidos orientan la forma en la que los alumnos obtienen sus calificaciones parciales así como su nota final. A su vez, la evaluación continua supone un seguimiento del aprendizaje del estudiante que facilita la detección de anomalías en sus calificaciones

en fases tempranas y permite la intervención inmediata para corregir la situación.

Normalmente, en los primeros cursos de estudios universitarios, el número de alumnos es elevado y ello redunda en el detrimento del control o seguimiento directo que se puede realizar a los estudiantes por parte del profesor. Por lo tanto, un sistema para la detección temprana de anomalías en la evaluación continua basado en técnicas de aprendizaje automático tendría la ventaja de ayudar al profesor a identificar qué alumnos pueden tener dificultades con la asignatura, de forma que puedan realizarse acciones individuales con ellos. Básicamente, el sistema aprendería de las experiencias del profesor (histórico de calificaciones de cursos anteriores) para identificar posibles problemas futuros.

Este artículo presenta la idea anterior, estructurando el contenido de la siguiente forma: en la sección 2 se detalla el esquema básico del sistema predictivo propuesto; en la sección 3 se explican los algoritmos de aprendizaje automático seleccionados; en la sección 4 se presentan los resultados experimentales; y finalmente, en la sección 5 se exponen las conclusiones y las ideas para trabajos futuros.

2. Sistema predictivo propuesto

En la figura 1 se muestra el esquema del método propuesto. Como se puede ver el método se divide en dos fases. En la primera fase se entrena el sistema predictivo utilizando el histórico de calificaciones de alumnos de cursos anteriores para la misma asignatura. La segunda fase se divide a su vez en dos tareas específicas:

1. En primer lugar se utiliza el sistema predictivo entrenado para obtener la nota esperada para los alumnos del curso actual.
2. A continuación se analizan las predicciones obtenidas para calcular la diferencia con las calificaciones reales de los alumnos.

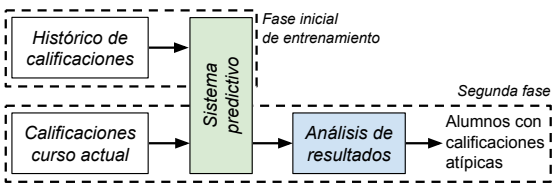


Figura 1: Esquema del método propuesto.

Con esta información se puede crear un sistema que notifique al profesor aquellos alumnos o alumnas cuyas calificaciones difieran de lo esperado un cierto umbral, seleccionando únicamente las que se consideren atípicas, es decir, aquellas que se encuentren por debajo de un percentil establecido (en nuestro sistema hemos utilizado el 10 %).

Los sistemas predictivos basados en aprendizaje automático requieren que se defina un conjunto de características que representen el problema al que se pretende dar solución. En el caso de estudio propuesto, las características básicas a tener en cuenta para cada alumno son: las calificaciones previas obtenidas, el grupo al que pertenece (mañana o tarde), y su género[1].

Cabe destacar que en el contexto de la evaluación de prácticas, la estimación del desempeño de un alumno en la primera práctica se tendrá que basar únicamente en su género y grupo, puesto que no se disponen todavía de calificaciones previas. Para la predicción de las siguientes prácticas sí que se podrán añadir los resultados obtenidos en las prácticas anteriores a este conjunto de características, por lo que a medida que avance el curso académico, el sistema predictivo tendrá más información de cada alumno y podrá realizar predicciones más precisas.

En la literatura existen multitud de algoritmos predictivos, por lo que resulta de especial interés seleccionar aquellos que minimicen el error tanto como sea posible, permitiendo optimizar la precisión de las estimaciones. En el siguiente apartado se describe la lista de algoritmos que se ha considerado, y posteriormente, en la sección de experimentación, se analizarán los resultados obtenidos por los mismos.

3. Algoritmos de predicción

En aprendizaje automático, los algoritmos que predicen un valor numérico continuo (en lugar de una categoría discreta) son conocidos como modelos de regresión. Por ello, hemos seleccionado una serie de algoritmos basados en diferentes estrategias para abarcar el mayor número de aproximaciones al problema y así poder evaluar su desempeño en el caso de estudio propuesto. Para esto hemos utilizado distintas herramientas y lenguajes de programación, incluyendo utilidades libres como WEKA, Python, Scikit-learn (Sklearn), XGBoost, LightGBM y Keras. A continuación se detalla la lista de algoritmos analizados, indicando la herramienta a la que pertenece y su versión.

En primer lugar se ha utilizado la herramienta de análisis de datos WEKA (v3.7.12) [11] para realizar los experimentos. En concreto se han considerado los siguientes métodos (el listado se muestra utilizando la nomenclatura y clasificación de la librería):

- Funciones:
  - GaussianProcesses [20]
  - IsotonicRegression [21]
  - LinearRegression [27]

[1] Institucionalmente no es posible consultar información adicional en el expediente del estudiante y, por lo tanto, no se pueden extraer ni utilizar más características.

  - LeastMedSq [23]
  - PaceRegression [26]
  - RBFNetwork [12]
  - RBFRegressor [12]
  - SMOreg [24]
- Árboles:
  - RandomForest [3]
  - M5P [25]
- Basado en prototipos (*lazy*):
  - LWL [1]
- Meta-algoritmos:
  - AdditiveRegression [13]
  - RandomSubSpace [14]
  - RandomCommittee [18]

También se han usado paquetes del lenguaje Python como Sklearn(0.19) [22], XGBoost(0.6), LightGBM(2.0.7) y Keras (2.0.8), para evaluar los siguientes algoritmos:

- Algoritmos lineales:
  - LinearRegression [27]
  - Ridge [15]
  - BayesianRidge [19]
- Árboles:
  - DecisionTree [10]
- Regla de los vecinos más cercanos:
  - KNeighbors [7]
- Boosting:
  - XGBoost XGB (xgboost) [4]
  - LightGBM LGBM (lightgbm) [17]
- Support Vector Machine (SVM):
  - SVR(SVM) [9]
- Artificial Neural Networks (ANN) con Keras [5]:
  - ANN-dense (capa 1, 32 neuronas)
  - ANN-avg (GlobalAveragePolling, dropout 0.1, 32 neuronas, dropout=0.2)

4. Experimentación

Los datos a evaluar han sido obtenidos a partir de las calificaciones de cuatro prácticas (P1, P2, P3 y P4) de 751 alumnos correspondientes a cuatro cursos académicos completos (2013-2016) de una asignatura de informática introductoria. El desglose de estos datos se puede ver en el Cuadro 1.

En el diagrama de cajas o de Whisker de la Figura 2 podemos apreciar visualmente algunos indicadores estadísticos sobre las calificaciones de estas prácticas, como la mediana, los cuartiles 1 y 3, y los valores considerados atípicos. Además, para mayor claridad, se han separado los valores atendiendo al grupo (mañana/tarde) y al género (masculino/femenino) de los alumnos. Observamos como las medianas de los alumnos del grupo de mañana son ligeramente superiores a los de la tarde.

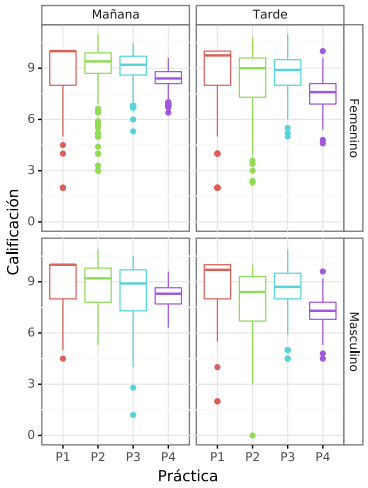


Figura 2: Distribuciones de las calificaciones dependiendo de las prácticas, el grupo y el género.

Para validar cada uno de los algoritmos de regresión seleccionados se ha utilizado la técnica de validación cruzada sobre 10 particiones (10-*fold cross-validation*, 10-CV en adelante) aplicada habitualmente para este tipo de tareas.

Para medir la calidad de los resultados obtenidos a partir de los algoritmos predictivos se ha utilizado el *error absoluto medio* (*Mean Absoluted Error*, referenciado como MAE en adelante). Se ha elegido esta métrica debido a su fácil interpretación en esta aplicación docente. Por ejemplo, si el MAE de un algoritmo es de $0,80$ puntos —dentro del sistema de referencia estándar en el ámbito docente de 0 a 10 puntos— significa que en promedio el modelo se equivoca en este valor para cada predicción que realice. Por lo que un modelo será mejor cuanto menor sea su MAE.

Según la calificación de la práctica que se quiera pre-

		Género		Totales
		Masculino	**Femenino**	
Grupo	**Mañana**	111	297	408 (54,3 %)
	Tarde	149	194	343 (45,7 %)
	Totales	260 (34,6 %)	491 (65,4 %)	751

Cuadro 1: Desglose del número de alumnos estudiados según grupo y género.

decir (P1, P2, P3 y P4) partiremos de un conjunto de datos inicial distinto. De esta forma establecemos cuatro escenarios o grupos de experimentos, como se indica a continuación:

1. Grupo, Género $\longrightarrow$ P1
2. Grupo, Género, P1 $\longrightarrow$ P2
3. Grupo, Género, P1, P2 $\longrightarrow$ P3
4. Grupo, Género, P1, P2, P3 $\longrightarrow$ P4

Cabe destacar que la importancia de los algoritmos utilizados se debe interpretar en orden cronológico inverso, dado que las predicciones de las últimas prácticas tienen una mayor retroalimentación debido a los resultados de las prácticas previas.

Como podemos observar en el Cuadro 2, cuando se incrementa el número de características, el error promedio decrece. Los algoritmos más precisos son aquellos que acumulan más resultados sombreados en verde, dado que significa que pertenecen al primer cuartil (mejores resultados). En el caso contrario se encuentran los algoritmos sombreados en rojo, que significa que pertenecen al último cuartil (peores resultados). Como cabía esperar, utilizar el histórico del alumno en la asignatura mejora significativamente la predicción de los sistemas.

Para contrastar los resultados obtenidos en el 10-CV hemos aplicado el test pareado de rangos con signo de Wilcoxon [28]. Este test de significancia es no paramétrico, lo que permite analizar distribuciones de datos que no se ajusten a la distribución normal.

Debido al gran número de algoritmos comparados se han dividido los resultados de los test estadísticos en dos grupos para mayor claridad: los pertenecientes al software WEKA y los implementados en Python. Los Cuadros 3 y 4 muestran estos resultados, respectivamente. Podemos ver como los algoritmos clásicos de regresión lineal múltiple tienen un buen comportamiento cuando su predicción se basa en pocas variables. Por otro lado, los algoritmos que han demostrado un mejor comportamiento promedio cuando el número de variables base se incrementa, corresponden a la familia de los llamados máquinas de vectores soporte (SMOreg y SVR), así como alguno de los basados en aumentado de gradiente extremo (XGB). Por el contrario, los que han demostrado un peor comportamiento contrastado corresponden a los basados en redes, como la red normalizada de función de base radial Gausiana (RBFNetwork) o redes neuronales con agrupación global de medias (ann-avg). Una posible explicación es que necesiten más muestras para poder ajustar mejor los modelos predictivos. También se observa este comportamiento en las técnicas de árboles de decisión basadas en un solo árbol (DecisionTree); sin embargo, cuando el algoritmo se basa en múltiples árboles de decisión (RandomForest) el resultado mejora.

Para probar los algoritmos con los datos descritos se ha creado un sistema interactivo online. Este sistema está disponible para su uso de forma pública en la dirección: **`https://goo.gl/hCTbJj`**. La implementación se ha realizado en Google Colaboratory usando un Notebook de Python, y permite seleccionar tanto el algoritmo a utilizar como la práctica a predecir. El propósito es ofrecer una versión del sistema que pueda ser probada de forma pública y además que se pueda adaptar fácilmente a otras asignaturas.

4.1. Fase de inferencia

Una vez determinado el algoritmo de predicción que mejor resultados obtiene (SVR según la experimentación anterior), solo resta aplicar el sistema propuesto para la predicción de las calificaciones en la asignatura. En nuestro caso de estudio, al constar de cuatro prácticas, se han entrenado cuatro modelos predictivos (uno por práctica). Para el entrenamiento de cada modelo se utilizó el histórico de los tres cursos anteriores (2013-2015), con un total de 461 calificaciones. La evaluación se realizó con el curso 2016 que constaba de 191 estudiantes.

Una vez disponemos de las calificaciones reales de cada una de las prácticas de los alumnos, procederíamos a calcular individualmente sus diferencias con respecto a las predicciones obtenidas mediante el modelo. Este proceso se realiza automáticamente, el profesorado únicamente debe establecer los umbrales (absolutos o relativos). En nuestro caso hemos seleccionado un pencentil del 10 % de los valores MAE más elevados. En el Cuadro 5 se muestra el umbral obtenido en función de este percentil así como el número de trabajos que el sistema recomendaría revisar con más detalle.

A modo de ejemplo, el Cuadro 6 muestra los resultados reales obtenidos por el sistema para la práctica 3. En esta tabla se listan los 19 alumnos seleccionados (anonimizados usando un número identificador), las características utilizadas para la estimación (Género, Grupo, P1 y P2), la nota real de la práctica (columna “P3 (Real)”), la predicción obtenida por el sistema (columna “P3 (Pred)”), y el MAE calculado. Conviene destacar que en este caso, todavía restaría una actividad por realizar (P4) por lo que el profesor aun tendría margen para intervenir personalmente con los alumnos correspondientes.

4.2. Discusión

La intención principal del presente trabajo es proporcionar una herramienta para ayudar al profesor y avisarle de aquellas situaciones que requieran una especial atención. Para ello es necesario, en primer lugar, identificar los mejores sistemas predictivos a fin de seleccionar el que tenga una mayor precisión. Y, en se-

gundo lugar, entrenar dicho sistema predictivo usando el histórico de evaluaciones de la asignatura a la que se va a aplicar. De esta forma se consigue adaptar el sistema a la propia asignatura y sus actividades, y que aprenda las particularidades de la misma para aumentar la precisión de las recomendaciones obtenidas.

Según Kahneman [16], el error cometido en la toma de decisiones por seres humanos —en nuestro caso las calificaciones de las prácticas— está formado por dos factores: el sesgo y el ruido. En nuestro estudio, el sesgo está minimizado por el uso de rúbricas en las correcciones, por lo que prácticamente el error obtenido por los modelos es debido al ruido. El ruido consiste en que ante la misma situación (las mismas características) las respuestas son diferentes (los valores de las prácticas a predecir). Por ello, este método es más efectivo según se tengan más alumnos evaluados y más información sobre el perfil de cada alumno. En este caso, únicamente disponemos del grupo, sexo y de las calificaciones en las prácticas que vamos evaluando. Es por ello que en la primera práctica existe mucho ruido (ya que hay pocas características), pero este ruido disminuye según el número de prácticas aumenta. Lo deseable sería disponer desde el principio del estudio de información suficiente del alumno sobre diferentes competencias. De esta forma se podría intentar disminuir el ruido en los datos e incrementar notablemente la precisión del modelo, y, por consiguiente, la precisión de las recomendaciones dadas al profesor.

Cabría destacar que este sistema no sería efectivo o no se podría utilizar en el caso de disponer de un número reducido de calificaciones de alumnos o al comenzar a impartir una nueva asignatura donde no disponemos de un histórico de calificaciones.

Paquete	**Algoritmo**	**P1**	**P2**	**P3**	**P4**	**Media**
weka.functions	GaussianProcesses	1,22	1,17	0,85	0,42	0,92
	IsotonicRegression	1,21	1,2	0,87	0,56	0,96
	LinearRegression(wk)	1,21	1,17	0,85	0,42	0,91
	LeastMedSq	1,06	1,13	0,85	0,42	0,87
	PaceRegression	1,21	1,16	0,85	0,42	0,91
	RBFNetwork	1,23	1,2	0,88	0,64	0,99
	RBFRegressor	1,22	1,16	0,84	0,43	0,91
	SMOreg	1,06	1,1	0,83	0,42	0,85
weka.trees	RandomForest	1,22	1,16	0,85	0,41	0,91
	M5P	1,21	1,17	0,85	0,42	0,91
weka.lazy	LWL	1,21	1,18	0,86	0,54	0,95
weka.meta	AdditiveRegression	1,22	1,17	0,83	0,43	0,91
	RandomSubSpace	1,22	1,17	0,85	0,43	0,92
	RandomCommittee	1,22	1,17	0,88	0,46	0,93
sklearn.linear_model	LinearRegression(sk)	1,22	1,16	0,85	0,42	0,91
	Ridge	1,22	1,17	0,85	0,42	0,92
	BayesianRidge	1,22	1,17	0,85	0,42	0,92
sklearn.tree	DecisionTree	1,22	1,17	0,92	0,51	0,96
sklearn.neighbors	KNeighbors	1,25	1,16	0,84	0,42	0,92
xgboost	XGB	1,22	1,15	0,82	0,41	0,90
sklearn.svm	SVR(SVM)	1,07	1,09	0,81	0,42	0,85
keras	ann-dense	1,19	1,19	0,88	0,44	0,93
	ann-avg	1,27	1,22	0,95	0,54	1,00
lightgbm	LGBM	1,22	1,17	0,83	0,46	0,92

Cuadro 2: Resultados de los errores absolutos medios (sobre 10 puntos) de los algoritmos evaluados con la técnica de validación cruzada (10 particiones). Sombreado en verde se muestra el primer cuartil y en rojo el último cuartil por columna. Un valor menor representa un mejor resultado.

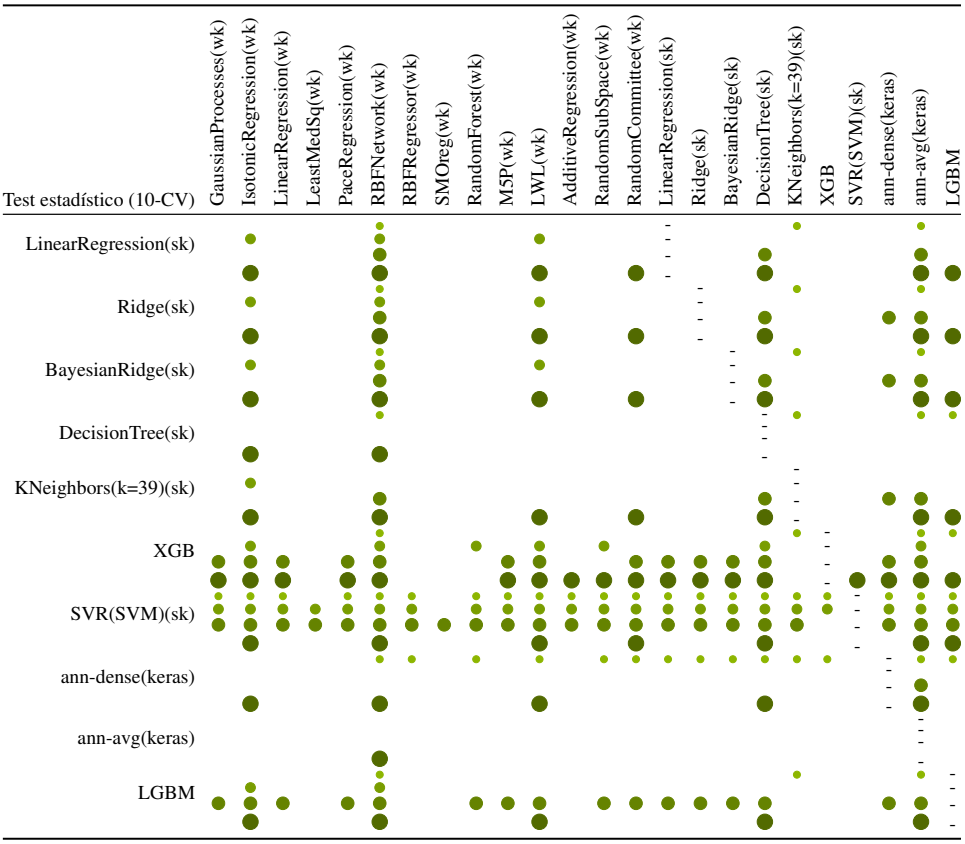
Cuadro 3: Test pareado de rangos con signo de Wilcoxon para los algoritmos del software WEKA. Se establece en un 95 % la significancia para determinar que los resultados del algoritmo de la fila son mejores que los de la columna. Las marcas ●, ●, ● y ● corresponden a los resultados positivos del test estadístico sobre los experimentos P1, P2, P3 y P4, respectivamente.

5. Conclusiones y trabajos futuros

En este trabajo se ha presentado un estudio de 24 algoritmos de inteligencia artificial, pertenecientes a diferentes categorías dentro del campo del aprendizaje automático, para la predicción de las calificaciones de los alumnos en una asignatura dada. Esta propuesta tiene como objetivo principal la detección temprana de anomalías en las calificaciones de los alumnos, para de esta forma asistir a la labor del profesor y ayudarle en el seguimiento de clases masificadas para que pueda localizar estos casos e intervenir a tiempo.

El sistema propuesto se ha evaluado usando datos de cuatro prácticas de 751 alumnos obtenidos de cuatro cursos académicos completos de una asignatura de informática introductoria. En esta experimentación se ha observado que los algoritmos tradicionales basados en regresión lineal múltiple obtienen buenos resultados cuando se utilizan pocas variables como base para la predicción. Cuando el número de variables aumenta, los algoritmos que obtienen mejores resultados son los correspondientes a la familia de los llamados máquina de vectores soporte (SVM) y los recientes algoritmos basados en el aumentado de gradiente extremo (XGB).

Posteriormente, el sistema propuesto se evaluó utilizando el mejor algoritmo determinado en la experimentación anterior (SVM). Para esto se realizó la predicción de las notas de los alumnos de la asignatura indicada, determinando que el sistema identificaba co-



Cuadro 4: Test pareado de rangos con signo de Wilcoxon para los algoritmos implementados en Python. Se establece en un 95 % la significancia para determinar que los resultados del algoritmo de la fila son mejores que los de la columna. Las marcas •, •, ● y ● corresponden a los resultados positivos del test estadístico sobre los experimentos P1, P2, P3 y P4, respectivamente.

Práctica	Umbral	Casos seleccionados
P1	1,9	11
P2	2,3	19
P3	1,8	19
P4	0,8	19

Cuadro 5: Umbral calculado para cada práctica usando un percentil del 10 % y número de casos seleccionados por el sistema en base a dicho umbral.

rrectamente los alumnos cuya nota se desviaba de la esperada, y notificando al profesor para que revise aproximadamente el 10 % del total de alumnos.

Como trabajo futuro se pretende ampliar el estudio a otras asignaturas del ámbito universitario o incluso a titulaciones completas. De esta forma se podría obtener un informe sobre anomalías producidas por alumnos según su historial, o bien, por las propias asignaturas.

Id	Género	Grupo	P1	P2	P3 (Real)	P3 (Pred)	MAE
624	F	M	10	9.3	5.3	9.2	3.9
715	F	T	2	7.6	5.2	8.6	3.4
632	F	M	10	10	6	9.3	3.3
676	F	T	6	7.3	9.8	6.9	2.9
679	F	T	6	7.3	9.8	6.9	2.9
628	M	M	10	8.9	6.4	9	2.6
659	F	M	10	9.9	6.7	9.3	2.6
675	F	T	6	6.1	9.8	7.2	2.6
583	M	M	10	8.7	6.6	9	2.4
636	M	M	10	9.5	6.7	9.1	2.4
663	M	T	6	6.4	9.1	6.9	2.2
593	M	M	8	7.7	9.8	7.7	2.1
681	M	T	8	5.7	9.8	7.7	2.1
577	M	M	8	7.8	9.7	7.7	2
600	M	M	8	7.8	9.7	7.7	2
696	M	T	8	7.1	9.6	7.6	2
564	F	M	6	6.6	9.3	7.4	1.9
702	M	T	10	6.4	6.6	8.5	1.9
701	F	T	10	9.9	7.6	9.4	1.8

Cuadro 6: Resultados reales generados por el asistente para la práctica 3. Se muestran los 19 casos seleccionados en base al umbral de 1,8 obtenido para un percentil del 10 %.

Agradecimientos

El presente trabajo contó con una ayuda del Programa de Redes-I^{3}CE de investigación en docencia universitaria del Instituto de Ciencias de la Educación de la Universidad de Alicante (convocatoria 2018-19). Ref.: REDES-I3CE-2018-4369

Referencias

[1] Atkeson, C., A. Moore y S. Schaal: *Locally weighted learning*. AI Review, 1996.

[2] Barnes, T., K. Boyer, I. Sharon, H. Hsiao, N. T. Le y S. Sosnovsky: *Preface for the Special Issue on AI-Supported Education in Computer Science*. Intl. Journal of Artificial Intelligence in Education, 27(1):1–4, 2017.

[3] Breiman, L.: *Random Forests*. Machine Learning, 45(1):5–32, 2001.

[4] Chen, T. y C. Guestrin: *XGBoost: A Scalable Tree Boosting System*. CoRR, abs/1603.02754, 2016.

[5] Chollet, F. *y cols.*: *Keras*. `https://github.com/keras-team/keras`, 2015.

[6] Costa, E. B, B. Fonseca, M. A. Santana, F. F. de Araújo y J. 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, 2017.

[7] Cover, T.M. y P.E. Hart: *Nearest neighbor pattern classification*. Information Theory, IEEE Transactions on, 13(1):21–27, 1967.

[8] Daud, A., N.R. Aljohani, R.A. Abbasi, M.D. Lytras, F. Abbas y J.S. Alowibdi: *Predicting Student Performance using Advanced Learning Analytics*. En *Proceedings of the 26th Intl. Conference on World Wide Web Companion*, páginas 415–421. Intl. World Wide Web Conferences Steering Committee, 2017.

[9] Drucker, H., C.J.C. Burges, L. Kaufman, A.J. Smola y V. Vapnik: *Support vector regression machines*. En *Advances in neural information processing systems*, páginas 155–161, 1997.

[10] Dumont, M., R. Marée, L. Wehenkel y P. Geurts: *Fast multi-class image annotation with random subwindows and multiple output randomized trees*. En *Proc. Intl. Conference on Computer Vision Theory and Applications (VISAPP)*, volumen 2, páginas 196–203, 2009.

[11] Eibe, F., M.A. Hall, I.H. Witten y J.C. Pal: *The WEKA workbench*. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques, 4, 2016.

[12] Frank, E.: *Fully Supervised Training of Gaussian Radial Basis Function Networks in WEKA*. Informe técnico 04/14, Department of Computer Science, University of Waikato, 2014.

[13] Friedman, J.H.: *Stochastic Gradient Boosting*. Informe técnico, Stanford University, 1999.

[14] Ho, T.K.: *The Random Subspace Method for Constructing Decision Forests*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8):832–844, 1998, ISSN 0162-8828.

[15] Hoerl, A.E. y R.W. Kennard: *Ridge regression: Biased estimation for nonorthogonal problems*. Technometrics, 12(1):55–67, 1970.

[16] Kahneman, D., A.M. Rosenfield, L. Gandhi y T. Blaser: *NOISE: How to overcome the high, hidden cost of inconsistent decision making*. Harvard business review, 94(10):38–46, 2016.

[17] Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye y T. Y. Liu: *LightGBM: A highly efficient gradient boosting decision tree*. En *Advances in Neural Information Processing Systems*, páginas 3149–3157, 2017.

[18] Lira, M.S., R.B. de Aquino, A. Ferreira, M.A. Carvalho, O.N. Neto y G.S.M. Santos: *Combining multiple artificial neural networks using random committee to decide upon electrical disturbance classification*. En *Neural Networks, 2007. IJCNN 2007. Intl. Joint Conference on*, páginas 2863–2868. IEEE, 2007.

[19] MacKay, D.J.C.: *Bayesian interpolation*. Neural computation, 4(3):415–447, 1992.

[20] Mackay, D.J.C.: *Introduction to Gaussian Processes*, 1998.

[21] Mair, P., K. Hornik y J. de Leeuw: *Isotone optimization in R: pool-adjacent-violators algorithm (PAVA) and active set methods*. Journal of statistical software, 32(5):1–24, 2009.

[22] Pedregosa, F., 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.

[23] Rousseeuw, P.J. y A.M. Leroy: *Robust regression and outlier detection*. 1987.

[24] Smola, A.J. y B. Schoelkopf: *A tutorial on support vector regression*. Informe técnico, 1998. NeuroCOLT2 NC2-TR-1998-030.

[25] Wang, Y. y I.H. Witten: *Induction of model trees for predicting continuous classes*. En *Poster papers of the 9th European Conference on Machine Learning*. Springer, 1997.

[26] Wang, Y. y I.H. Witten: *Modeling for optimal probability prediction*. En *Proceedings of the Nineteenth Intl. Conference in Machine Learning*, páginas 650–657, Sydney, Australia, 2002.

[27] Weisberg, S.: *Applied linear regression*, volumen 528. John Wiley & Sons, 2005.

[28] Wilcoxon, F.: *Individual comparisons by ranking methods*. Biometrics bulletin, 1(6):80–83, 1945.